

SQL DIAGNOSTIC MANAGER 11.0

I D  R A

Table of Contents

24x7 SQL performance monitoring, alerting, and diagnostics	24
Access the SQL Diagnostic Manager Release Notes	25
New features and fixed issues	26
11.0 New Features	26
11.0 Fixed Issues.....	26
Known issues.....	30
Known Issues in version 11.0.....	30
Previous known issues.....	30
Known issues in version 10.6.....	30
Known issues in version 10.5.1	31
Known Issues in version 10.5.....	31
Known Issues in version 10.4.....	31
Known Issues 10.3.....	32
Known Issues in version 10.2.3.....	33
Known issues in version 10.2.2.....	33
Known issues in version 10.2.1	33
Known issues in version 10.1.4.....	34
Known issues in version 10.1.1	34
Known issues in version 10.1.....	35
Known issues in version 10.0.....	35
Known issues in version 9.1.....	35
Known issues in version 9.0.....	37
Known issues in version 8.6.....	38
Known issues in version 8.5.....	39
Known issues in version 8.0.....	40
Known issues in version 7.5.....	40
Known issues in versions previous to 7.5	43
Recommended IDERA Solutions	45
Get started.....	46
Upgrade to this release.....	47
Upgrade from SQL Diagnostic Manager 5.x or higher to SQL Diagnostic Manager 10.6 checklist.....	47
Upgrade SQL Diagnostic Manager in a clustered environment.....	48

Perform a standard upgrade	49
Upgrade SQLDM console only	50
Test your deployment.....	51
Installation and deployment.....	52
Get ready for your SQL Diagnostic Manager and IDERA Dashboard deployment	52
Get ready for your SQLDM Mobile or IDERA Newsfeed deployment	52
Get ready for your SQL Diagnostic Manager Console only deployment	52
Supported installation scenarios	53
Choose your deployment strategy.....	55
Deploy the IDERA Dashboard and SQL Diagnostic Manager.....	56
Deploy SQLDM Mobile or IDERA Newsfeed	87
How to install SQL Diagnostic Manager Console only	99
Deploy SQL Diagnostic Manager in a high availability group	100
How SQL Diagnostic Manager works with a virtual environment	106
Trial installation instructions	112
Configure your deployment	119
Connect to the SQL Diagnostic Manager Repository	120
Add your SQL Server instances.....	121
Configure your virtual machine connections	146
Configure console options.....	151
Set your SQL Diagnostic Manager Repository grooming schedule	158
Use custom counters to track metrics	161
Work with tags.....	189
User Creation Scripts	193
Welcome to SQL Diagnostic Manager	198
What is SQL Diagnostic Manager?	199
What is the IDERA Dashboard?	199
What is SQL Diagnostic Manager Mobile?	199
How does SQL Diagnostic Manager help me?	200
Use the SQL Diagnostic Manager interface.....	201
Navigation pane	202
Refresh and pause views	203
Charts.....	204
Monitor multiple SQL Server instances	206

Server Configuration Properties	209
Use the SQL Diagnostic Manager web console.....	215
Find answers	216
View help topics	216
Print help topics	216
Customer support portal	217
Access the IDERA Customer Support Portal	217
Support Status	218
Copy of Support Status.....	219
Definitions	220
Navigate the IDERA Dashboard	222
Register SQL Diagnostic Manager in the IDERA Dashboard	223
Edit a Product.....	223
Delete a Product.....	223
Migrate Product.....	223
Register SQL Diagnostic Manager	224
Navigate the web console dashboard.....	226
Additional options	226
Search	226
Heatmap Calculation Configuration	226
Refresh.....	226
Use views to get an overview of your environment	227
By Status.....	227
By Tags.....	227
By Most Critical Instances.....	227
Use sub-views to identify and diagnose issues	228
List sub-view.....	228
Thumbnail sub-view	230
Heatmap sub-view	233
Navigate the web console single instance dashboard.....	237
History Range Control.....	238
Alert and Health Summary	238
Prescriptive Analysis Summary	238

Server Properties	238
Prescriptive Analysis	239
Alert Timeline	240
Alert Categories.....	240
Charts.....	240
Category views	241
Access the single instance dashboard	242
View your SQL Server sessions information	243
View your SQL Server queries information	245
View your SQL Server query waits information	275
View your SQL Server resources information	276
View your SQL Server databases information	281
View your SQL Server alerts.....	284
View your top values in the web console	287
Records	289
Top servers by alerts	289
Top servers by CPU usage.....	289
Top servers by memory usage.....	289
Top Servers by response time	289
Top Servers by waits	290
Top servers by queries	290
Top servers by I/O	290
Top servers by sessions	290
Top servers by blocked sessions	290
Top Servers by active connections	291
Top servers by disk space utilization	291
Top servers by tempdb utilization	291
Top sessions by CPU usage	291
Top queries by execution time	292
Top databases by size.....	292
Top databases by growth (Last 7 days).....	292
Top databases by alerts.....	292
Top databases by activity	293
View all your alerts in the web console	294
Organize your alerts.....	294

Additional options	295
Access alert details.....	297
Advanced Filter	299
Create custom dashboards	301
Add a dashboard	301
Add widgets to your dashboard	301
Actions to perform in your Custom Dashboards View	302
Filter results by time frame.....	302
Edit your dashboard	302
Remove a dashboard	303
Duplicate a dashboard.....	303
Custom dashboards widget types.....	304
Navigate the IDERA Newsfeed and SQLDM Mobile.....	305
Configure your Newsfeed and Mobile deployment.....	306
Suggested configuration workflow	306
Available configuration tasks	306
Assign an owner to a server	308
Assign an owner to an application	309
Change a server or an application profile.....	310
Change the Web Server settings for SQLDM Mobile	311
Change your account settings.....	312
Change the Newsfeed administrator	313
Configure how the IDERA Newsfeed Platform authenticates with SQLDM.....	314
Configure the IDERA Newsfeed Repository.....	315
Configure SQLDM Mobile and Newsfeed email settings	319
Invite coworkers.....	320
Manage users.....	321
Manage your profile	323
View application status.....	324
Create an account and Sign Up	325
Sign up via the SQLDM Console.....	325
Sign up via SQLDM Mobile	325
IDERA Newsfeed and SQLDM Mobile Log On	327
Navigate the IDERA Newsfeed Interface	328

Activity Timeline.....	328
Application	329
Comment.....	329
Coworker	330
Followers	330
Why do I see my own posts in my feed?	331
How do I follow someone or something?	331
How do I see who is following me?	331
How do I review the things and people I am following?	331
How do I follow specific stories?	331
Like or Unlike.....	331
Mention.....	331
My Bookmarks.....	332
My Directory	332
My Favorites.....	333
My Most Recent feed	333
My Notifications	334
Post.....	335
Profiles.....	335
Server.....	336
Status Updates.....	337
Story.....	337
Walls.....	337
Navigate SQLDM Mobile Dashboards.....	340
Active alerts	340
Database details and databases dashboard	340
Home Page	341
Key resource metrics	341
My Servers	343
SQLDM Dashboard	344
Sessions.....	344
SQL Server Agent.....	346
Customize your experience	347
Add servers and coworkers to the SQLDM Mobile Home Page.....	347
Change who and what you follow	347

Change your Notification Settings	347
Tips and tricks	350
Announce maintenance plans.....	350
Ignore specific stories	350
Search for specific stories.....	351
Inform your coworkers about your server status	351
Use bookmarks to track critical servers.....	351
Kill a session remotely	351
Run queries to fix issues on remote servers	351
Start or stop a service remotely	353
Monitor SQL Server Performance.....	354
See the general status of all your monitored SQL Server instances	354
Get the configuration properties of all your monitored SQL Server instances.....	354
Get an overview of a group of SQL Server instances.....	354
Get an overview of a specific SQL Server instance	354
Interact with the selected SQL Server instance using IDERA Newsfeed	354
Launch SQL Workload Analysis	355
Launch SQL Workload Analysis	355
Monitor instances.....	356
Access the Overview tab	356
Navigate to your SQL Server instances.....	357
Get the SQL Server performance overview.....	361
Get the SQL Server performance details.....	392
Monitor SQL Server configuration settings.....	394
View active alerts for this instance.....	396
View past performance	400
Pause alerts and decrease collection	405
Configure SQL Diagnostic Manager monitoring properties.....	406
11.0 Copy Dashboard for Azure SQL.....	457
Monitor sessions	461
Access the Sessions tab	461
Alerts associated with the Sessions tab.....	461
Get sessions performance summary.....	463
Get sessions performance details	465

Analyze locked sessions.....	469
Analyze blocked sessions	471
Monitor query performance	474
Access the Queries tab.....	474
Use the Queries tab.....	474
Understand the difference between Signature Mode and Statement Mode.....	475
Use Filter Settings to customize your view.....	475
View the query monitor signature mode	476
View the query monitor statement mode.....	478
View the query history	480
View query waits	482
Query Details	483
Monitor resources	487
Access the Resources tab.....	487
Alerts associated with the Resources tab	487
Get the resource performance summary.....	490
Get CPU performance details	493
Get memory performance details	494
Get disk performance details	496
Get disk size details.....	498
View file activity	499
Manage your procedure cache.....	501
View server waits.....	502
Monitor databases	503
Percentage of Database/Log Full	503
Potential Maximum Available Space.....	503
Available Expansion Space	503
Access the Databases tab	503
Alerts associated with the Databases tab.....	504
Get the database status summary	506
Monitor AlwaysOn Availability Groups.....	508
Get the tempdb status summary	512
Monitor database configuration settings	514
View database status	515
Analyze database files	517

Analyze backups and restores	522
Analyze tables and indexes	523
Monitor database mirroring	527
Monitor services	531
Access the Services tab	531
Alerts associated with the Services tab	531
Get an overview of your services performance	534
Monitor SQL Agent jobs	535
Monitor full text searches	536
Analyze replication information.....	537
Monitor logs.....	539
Access the Logs tab	539
Available actions	539
Filter SQL Server events.....	541
Search for SQL Server events	542
Cycle the SQL Server log.....	543
Configure logs	544
Analyze your SQL Server instance.....	545
Access the Analyze tab.....	545
Run a prescriptive analysis on your SQL Server	547
Configure prescriptive analysis options	550
Schedule analysis.....	559
Fine-tune your analysis.....	560
Identify a problem and optimize scripts.....	565
View your prescriptive analysis results	566
Diagnose queries.....	567
View the Timeline for this instance	568
Access the Timeline view	568
Access more alert detail from the Timeline	568
DBaaS custom dashboard	569
Azure SQL Monitoring	570
Azure Discovery	571
Azure Overview.....	573
Dashboard	573

Details	573
Active Alerts	573
Timeline	574
Azure Sessions	575
Summary	575
Details	575
Locks	575
Blocking	575
Azure Resources	577
Summary	577
CPU	577
Memory	577
Disk	578
Server Waits	578
Azure Databases	579
Summary	579
Configuration	579
Azure Analysis	580
Organize alerts	581
Organize your alerts by Active Alerts	581
Organize your alerts by the Current View	581
Active	581
By Severity	581
By Server	581
By Metric	582
Agent Job Failures	582
Blocked Processes	582
Oldest Open Transactions	582
Query Monitor	582
Table Reorganization	582
Custom	582
Organize your alerts using the Filter Options	582
Alert on SQL Server Metrics	583
Database- or disk-level alerts	583

Alert templates.....	583
Informational alerts	584
IDERA Newsfeed	584
SQLDM Mobile	584
Configure alerts.....	585
Alert refresh	586
Table Fragmentation alert thresholds.....	586
Apply different alert thresholds to multiple databases or disks	586
Alert templates.....	586
Informational alerts	586
Configure alert templates.....	588
Configure alert template settings	594
Configure database thresholds	597
Configure disk thresholds.....	599
Set advanced alert options.....	600
Configure alerts for important cluster events	620
Configure how SQL Diagnostic Manager responds to alerts.....	621
Access the Alert Actions and Responses window	621
Available Alert Responses tab actions	621
Available Action Providers tab actions	623
Configure automated responses to alerts	625
Import an alert response	638
Export an alert response.....	640
Report on SQL Server Performance	641
Customize reports.....	641
Deploy reports to Microsoft Reporting Server	641
Report filters.....	642
Case Insensitive.....	642
Chart Type	642
Compare Server	642
Compare with same server	642
Database, Database Name	642
Forecast Type, Forecast Units	642
Drive Name	642

End Hour	642
Include Batches/SQL Statements/Stored Procedures/System Databases.....	642
Metric	643
Min. Duration	643
Min. CPU/Executions/Reads/Writes	643
Min. Bytes Read/Bytes Written/Data Size/Growth/Transactions	643
Number of Applications	643
Number of Databases/Servers	643
Number of Queries	643
Action	643
Repository User	643
Workstation	643
Workstation User.....	643
Period	644
Range	644
Sample.....	644
Search Item and Search Value.....	644
Server.....	644
Show Problems Only.....	644
Signature Mode	644
Start Time, Start Hour.....	644
Tag	644
Monitor reports	645
Access the Reports view	645
Server Monitor Reports.....	645
Virtualization Monitor Reports	646
Activity Monitor Reports	646
Server monitor reports	647
Virtualization monitor reports.....	658
Activity monitor reports.....	661
Analysis reports.....	663
Access the Reports view	663
Server Analysis Reports	663
Database Analysis Reports	664
Resource Analysis Reports.....	664

Server analysis reports	666
Database analysis reports	677
Resource analysis reports.....	687
Plan reports	697
Access the Reports view	697
Available reports	697
Disk Space Usage Forecast.....	698
Database Growth Forecast	699
Table Growth Forecast.....	700
Deploy SQL Diagnostic Manager reports to Reporting Services.....	701
Access the Reports Deployment wizard.....	701
Select reports to deploy	702
Specify a reporting services server and data source.....	706
Configure report email notifications.....	707
View the report deployment summary	708
Custom reports	709
Access Custom Reports.....	709
Create a custom report	711
Import a custom report	720
Export a custom report.....	721
Manage SQL Diagnostic Manager	722
Use application security to manage SQL Diagnostic Manager	723
Enable Application Security	723
Available actions on the Application Security window.....	723
Add users to SQL Diagnostic Manager	725
Edit user permissions.....	733
Use the Change Log to review changes in your SQL Diagnostic Manager configuration	735
Access the Change Log view	735
Change Log Window Properties	735
Available actions	735
Available columns	736
Use the SQL Diagnostic Manager import/export wizard.....	737
Access the import/export wizard	737
Use PowerShell to automate SQL Diagnostic Manager functions.....	738

About PowerShell	738
Install the SQL Diagnostic Manager provider for PowerShell.....	738
Applications available to use with PowerShell	739
PowerShell Cmdlets for SQL Diagnostic Manager.....	741
Customize your Monitored SQL Server properties within PowerShell	748
Manage your SQL Diagnostic Manager license	758
Use the trial license.....	759
Add a new license.....	760
Generate a new license key	761
Manage your SQL Server instances	763
Edit your Management Service configuration	764
Access the Management Service Configuration wizard	764
Select a SQL Server instance for the Management Service	765
Select a database for the Management Service	766
Monitored metrics.....	767
Batches Received (Per Second).....	768
Reduce the number and impact of T-SQL batches	768
Blocked Sessions (Count)	769
Buffer Cache Hit Ratio (Percent)	770
Reduce the percentage of memory used by SQL Server.....	770
Client Computers	771
SQL Compilations (Per Second)	772
Data Used (Percent)	773
Reduce the percentage of data used	773
Database Status	774
Disk Queue Length	775
Reasons why this metric has no value	775
Disk Time (Percent).....	776
Reduce disk time.....	776
I_O Waits.....	777
Lock Waits.....	778
Log Flushes (Per Second).....	779
Reduce log flushes	779

Memory Page Faults (Per Second)	780
Reduce memory page faults.....	780
Memory Paging	781
Reasons why this metric has no value	781
Memory Waits.....	782
Other Waits.....	783
Packets Received (Per Second)	784
Packets Sent and Received.....	785
Packets Sent (Per Second).....	786
Page Reads (Per Second)	787
Reduce page reads.....	787
Page Writes (Per Second).....	788
Reduce page writes.....	788
Percent of SQL Server Data Space Used	789
Percent of SQL Server Log Space Used	790
Physical Page Reads and Writes.....	791
Processor Queue Length.....	792
Re-compiles (Per Second)	793
Signal waits	794
SQL Server Batches Received (Per Second).....	795
SQL Server CPU Usage (Percent).....	796
Lower SQL Server CPU usage	796
Log Space Used (Percent).....	797
Reduce the percentage of the log used	797
SQL Server Memory Usage.....	798
SQL Server Response Time (ms).....	799
Total Memory Usage	800
Transaction Log Waits.....	801
Transactions (Per Second).....	802
Wait type category - Backup.....	803
Wait type category - I/O Page Latch.....	804
Wait type category - Lock	805

Wait type category - Memory.....	806
Wait type category - Non-I/O Page Latch.....	807
Wait type category - Non-Page Latch.....	808
Wait type category - Other.....	809
CXPACKET wait type actions	818
LOGMGR_RESERVE_APPEND.....	819
SOS_STACKSTORE_INIT_MUTEX	820
THREADPOOL	821
Wait type category - Transaction Log	822
LOGBUFFER.....	823
Metric alerts.....	824
Category: Backup	824
Category: Databases	824
Category: DBaaS.....	826
Category: Logs.....	827
Category: Operational	827
Category: Queries.....	828
Category: Resources	828
Category: Services.....	829
Category: Sessions	830
Category: Virtualization	831
Availability Group Estimated Data Loss (Seconds) alert.....	832
Availability Group Estimated Recovery time (Seconds) alert.....	833
Availability Group Log Send Queue Size (KB) alert	834
Availability Group Preferred Replica alert	835
Availability Group Redo Queue Size (KB) alert	836
Availability Group Redo Rate (KB\sec) alert	837
Availability Group Role Change alert	838
Availability Group Synchronization Health alert.....	839
Availability Group Synchronization Performance (Seconds) alert.....	840
Average Data IO Percent High	841

Average Data IO Percent Low	842
Average Disk Milliseconds Per Read alert	843
Average Disk Milliseconds Per Transfer alert.....	844
Average Disk Milliseconds Per Write alert.....	845
Average Log Write Percent High	846
Average Log Write Percent Low	847
Blocked Sessions (Count) alert.....	848
Reduce the number of blocked sessions	848
Blocking Session Wait Time (Seconds) alert.....	850
Blocking Sessions Report	850
Reduce blocked sessions wait time	851
Client Computers (Count) alert	852
CLR Enabled alert.....	853
Cluster Active Node alert	854
Cluster Failover alert.....	855
Prevent cluster failover alerts when you have a scheduled failover	855
CPU Credit Balance	856
CPU Credit Usage	857
Database Average Memory Usage Percent High	858
Database Average Memory Usage Percent Low	859
Database Full (Percent) alert.....	860
Calculate the database full percentage using the autogrow settings.....	860
Database Full (Size) alert	861
Database Read/Write Error Occurred alert.....	862
Database Status alert.....	863
Reduce the number of database status issues	863
Data File Autogrow alert	864
Days since last backup alerts.....	865
Days since last backup at the instance-level	865
Days since last backup per-database.....	865
Days since last backup for availability groups.....	866
Deadlock alert	867

Resolve deadlocks	867
Disk Queue Depth.....	868
Disk Reads Per Second alert.....	869
Disk Transfers Per Second alert	870
Disk Writes Per Second alert.....	871
Distribution Latency (Seconds) alert	872
Causes of high distribution latency.....	872
Reduce distribution latency	872
DTC Status alert	873
Filegroup Space Full (Percent) alert.....	874
Calculate the filegroup space full percentage using the autogrow settings	874
Filegroup Space Full (Size) alert.....	876
Full-Text Search Status alert	877
Host CPU Usage (Percent) alert.....	878
Reduce your host CPU usage.....	878
Host Memory Swap Detected alert	879
Host Memory Usage (Percent) alert	880
Reduce your host memory usage.....	880
Host Power State alert.....	881
Index Row Hits (Percent) alert.....	882
In-Memory Storage Usage Percent High.....	883
In-Memory Storage Usage Percent Low.....	884
Last Full-Text Catalog Update (Hours) alert	885
Latent replication transaction (Count)	886
Log File Autogrow alert.....	887
Log Full (Percent) alert	888
Calculate the log full percentage using the autogrow settings	888
Log Full (Size) alert.....	889
Longest Running Version Store Transaction (Minutes) alert	890
Maintenance Mode Enabled alert	891
Max Session Percent High.....	892
Max Session Percent Low	893

Max Worker Percent High	894
Max Worker Percent Low	895
Mirror Commit Overhead (Milliseconds) alert	896
Mirrored Server Role Change alert	897
Mirroring Oldest Unsent Transactions (Minutes) alert.....	898
Mirroring Preferred Configuration alert.....	899
Mirroring Status alert.....	900
Mirroring Unrestored Log (KB) alert.....	901
Mirroring Unsent Log (KB) alert.....	902
Mirroring Witness Connection alert	903
Oldest Open Transaction (Minutes) alert.....	904
OLE Automation Disabled alert.....	905
OS Average Disk Queue Length (Count) alert	906
Reduce queue length	906
OS Average Disk Queue Length Per Disk (Count) alert.....	907
Reduce queue length	907
OS Disk Free Space (Size) alert.....	908
OS Disk Full (Percent) alert.....	909
OS Disk Time (Percent) alert.....	910
Reduce OS processor time.....	910
OS Disk Time Per Disk (Percent) alert	911
Reduce OS processor time.....	911
OS Memory Usage (Percent) alert	912
Reduce OS memory usage.....	912
OS Metrics Collection Status alert.....	913
OS Paging (Per Second) alert	914
Reduce memory paging.....	914
OS Privileged Time (Percent) alert.....	915
OS Processor Queue Length (Count) alert.....	916
Reduce your OS processor queue length.....	916
OS Processor Time (Percent) alert	917
Reduce OS processor time.....	917

Reasons why the CPU Usage metric has no value	917
OS User Time (Percent) alert	918
Page Life Expectancy alert.....	919
Diagnose problems with your SQL Server instance	919
Procedure Cache Hit Ratio (Percent) alert.....	920
Manage your Procedure Cache Hit Ratio	920
Query Monitor Events (Count) alert	921
Read Latency	922
Read Throughput	923
Service Tier Changes.....	924
Session CPU Time (Seconds) alert	925
Session Tempdb Space Usage (MB) alert	926
SQLdm Repository Grooming Timed Out alert.....	927
SQLDM Service Status alert	928
SQL Server Active Directory Helper Service Status alert.....	929
SQL Server Agent Job Completion alert	930
SQL Server Agent Job Failure alert	931
SQL Server Agent Log alert	932
Use a size limit on your SQL Server Agent log	932
SQL Server Agent Long Running Job (Minutes) alert	933
SQL Server Agent Long Running Job (Percent) alert.....	934
SQL Server Agent Status alert	935
SQL Server Agent XPs Disabled alert.....	936
SQL Server Browser Service Status alert	937
SQL Server CPU Usage (Percent) alert	938
Reduce your SQL Server CPU usage.....	938
SQL Server Data Used (Percent) alert	939
Reduce your SQL Server database space used.....	939
SQL Server Error Log alert	940
Use a size limit on your SQL Server Error log.....	940
SQL Server Log Used (Percent) alert	941
Reduce your SQL Server log space used	941

SQL Server Memory Usage (Percent) alert.....	942
Reduce the percentage of memory used by SQL Server	942
SQL Server Response Time (Milliseconds) alert	944
Reduce your SQL Server response time.....	944
SQL Server Service	945
Swap Usage	946
Table Fragmentation (Percent) alert	947
Tempdb Contention (ms) alert.....	948
Unsubscribed Transactions (Count) alert.....	949
Unsubscribed Transactions (Seconds) alert.....	950
User Connections (Percent) alert	951
Version Store Generation Ratio alert	952
Version Store Size (MB) alert	953
VM CPU Ready Wait Time (ms) alert.....	954
VM CPU Usage (Percent) alert.....	955
Reduce the percentage of available virtual CPU time used by your VM.....	955
VM Host Server Change alert	956
VM Memory Swap Delay Detected alert	957
VM Memory Usage (Percent) alert	958
Reduce the percentage of memory used by SQL Server	958
VM Power State alert.....	959
VM Reclaimed/Ballooned Memory (KB) alert	960
Reduce the percentage of memory used by SQL Server	960
VM Resource Configuration Change alert	961
VM Resource Limits Detected alert	962
WMI Service Unavailable alert.....	963
Write Latency.....	964
Write Latency Low.....	965
Write Throughput.....	966
Advanced solutions.....	967
Migrate or Recover Your Installation.....	968
System diagnostics.....	969

Choose to migrate or recover SQL Diagnostic Manager	971
Create your migration or recovery plan.....	972
Migrate your SQL Diagnostic Manager installation	974
Recovering your SQL Diagnostic Manager installation	976
Migrate the SQL Diagnostic Manager Repository	977
Migrate the SQL Diagnostic Manager services.....	979
Deploy SQL Diagnostic Manager in a clustered environment	980
Windows Server 2003 clustered environment.....	981
Windows Server 2008 clustered environment.....	984
Windows Server 2012+ clustered environment.....	987
Deploy IDERA Newsfeed in a clustered environment.....	993
Deploy the IDERA Newsfeed to a Windows 2003 cluster.....	994
Deploy the IDERA Newsfeed to a Windows 2008 cluster.....	997
Deploy the IDERA Newsfeed to a Windows 2012 cluster.....	999
Default alert IDs.....	1001
Troubleshoot WMI connectivity issues	1017
Using WbemTest (Windows Management Instrumentation Tester)	1017
Error: The RPC Server Is Unavailable	1017
Error: Access Denied	1018
Integrate SQL Diagnostic Manager with SCOM.....	1020
Understanding the SQLDM Management Pack health model	1020
Install and configure the SQLDM Management Pack for SCOM.....	1021
View SQL Diagnostic Manager details in SCOM	1022

24x7 SQL performance monitoring, alerting, and diagnostics

- Performance monitoring for physical, virtual, and cloud environments
- Monitor queries and query plans to see the causes of blocks and deadlocks
- Adaptive & automated alerting with 100+ pre-defined and configurable alerts
- Capacity planning to see database growth trends and minimize server sprawl
- Monitor application transactions with SQL Workload Analysis add-on
- View expert recommendations from SQL Doctor to optimize performance
- View summary of top issues and alerts with the web console add-ons

Access the SQL Diagnostic Manager Release Notes

IDERA SQL Diagnostic Manager provides an unprecedented level of diagnostic information on the health, performance, and status of SQL Server instances across your environment. You can view, diagnose, and report on critical performance statistics from a central point of control.

Using a unique agent-less architecture, SQL Diagnostic Manager gathers diagnostic information in real time, keeping database administrators and managers informed by providing customized alerts. Easy to install and use, SQL Diagnostic Manager is an indispensable SQL Server management tool that benefits both experienced and junior database administrators.

To get a glimpse into the newest features, fixed issues, and known issues in this release of SQL Diagnostic Manager, review the following sections of the Release Notes:

- [See new features in this release](#)
- [Review issues fixed by this release](#)
- [Review previous features and fixed issues](#)
- [See known issues](#)
- [See list of recommended IDERA Solutions](#)

To see the summary of the Support Status for SQL Diagnostic Manager click [here](#).

New features and fixed issues

SQL Diagnostic Manager provides the following new features and fixed issues.

i Customers that have installed SQLDM 11.0.0.3871 build and want to install the new build 11.0.0.3886 will require to uninstall and reinstall. If you do not need Azure SQL SQL or Managed Instance support, there is no reason to uninstall and reinstall.

11.0 New Features

- SQL Diagnostic Manager includes full support for Azure SQL. This feature includes the following resources for the Azure SQL database:
 - The compatibility for *Azure SQL Managed Instance* with on-premises.
 - *Elastic pools* allow you to manage and scale multiple databases.
 - *Single databases* typically are used for development and small apps.
- This release includes the completion of Azure SQL alerts that were added in previous releases. Among other features, these alerts include:
 - New alerts notify if the allocated resources are more than needed.
 - New displays for newly collected metrics.
- New alert templates with revised thresholds and priority/rank specifically for DBaaS (Azure SQL and Amazon RDS)
- This version includes a new server wizard that uses Azure profile credentials, allowing you to discover other Azure DB assets.
- Azure profile credentials collect metrics from Azure monitor to extend SQL Diagnostic Manager.
- Prescriptive Analytics improvements, including new recommendations for Azure SQL DBaaS and Amazon RDS.

11.0 Fixed Issues

Alerts

SQLDM-29184 SQL Diagnostic Manager now correctly alerts table fragmentation based on the table size configuration.

SQLDM-30421 The advanced Alert Filter for the "Longest Running version Store Transaction (Minutes)" alert is working as configured.

SQLDM-30487 The 'Clear Alert' option in the 'Availability Group Role Change' alert has been restored.

SQLDM-30528 SQLDM shows "Recommendations are available" operational alert in the Active Alert list.

SQLDM-30788 SQLDM Disk space alerts are now correctly triggering after an upgrade.

SQLDM-30825 The Disk Full alerts are no longer triggered for excluded disks in the alerts configuration.

SQLDM-30963 SQL Diagnostic Manager now correctly shows the correct alerts and thresholds for page life expectancy metric when using the Critical-Only Template.

SQLDM-31039 The "Show Historical View" button in the Alerts Tab is now working as expected.

Alert Response

SQLDM-29208 The 'Enable the Query Wait' Alert Action will properly execute when an Alert Response is triggered.

Availability Groups

SQLDM-30083 SQL Diagnostic Manager no longer encounters an error when removing an instance that happens to be a replica within an availability group.

SQLDM-30197 SQLDM now discovers preferred replica information from Availability Group settings.

Cloud Monitoring

SQLDM-30344 The notifications for blocked sessions and SQL Agent Job Failures when monitoring RDS instances are now working correctly.

Custom Counters

SQLDM-30385 The newly created custom counters are added to the existing templates and can be applied to new servers.

SQLDM-30524 The Custom Counter alerts when using T-SQL script are raising correctly.

Database

SQLDM-30575 The Database status is now displaying accurate status.

Extended Events

SQLDM-29005 SQLDM now correctly add new servers and allows to select 'Extended Events' as the collection method for the Activity Monitor.

SQLDM-29033 The performance of the query that collects Replication Statistics has been improved.

General

SQLDM-29384 SQL Diagnostic Manager now correctly monitors its own repository.

SQLDM-29408 SQLDM installation is now completing successfully when the service account credentials contain special characters.

SQLDM-30114 SQL Diagnostic Manager now correctly reports the Publisher Names for SQL Server 2016 SP2 or later.

SQLDM-30364 Desktop Console Views now correctly keeps the custom views.

SQLDM-30453 IDERA Newsfeed is correctly configured after installing SQL Diagnostic Manager.

SQLDM-30480 SQL Diagnostic Manager now correctly displays baselines.

SQLDM-30493 SQLDM Desktop console now correctly shows the critical SQL Server Service status.

SQLDM-30770 SQL Diagnostic Manager successfully checks SQL Browser Service Alert status when the Remote Administrative connection is disabled and there is only a default SQL Server instance installed.

SQLDM-30936 SQL Diagnostic Manager displays the correct information when you select a secondary tab.

SQLDM-31025 The pause button now correctly works and it is integrated with the refresh button.

SQLDM-31042 SQL Diagnostic Manager now correctly displays the "Session Details" grid.

SQLDM-31068 The Query History view under the Queries tab is working as expected.

SQLDM-31070 The "View Session Details" option is now correctly working.

SQLDM-31071 The "View Locks" option is now correctly working.

Graphs

SQLDM-29069 The 'Disk busy per disk' metric graph now correctly loads under the Disk option from the Resources tab.

SQLDM-30037 The blocking displayed in the reports and in the graphs are synced correctly.

SQLDM-30137 The Alert Configuration window is no longer displaying inaccurate threshold graphs.

Hyper-V

SQLDM-28327 SQL DM successfully connects to Hyper-V Host Configuration.

IDERA Dashboard

SQLDM-30032 SQLDM web console is no longer reporting FIPS compliant environment errors.

SQLDM-30260 The IDERA Dashboard login information is now correctly encrypted.

SQLDM-30350 The SQLDM Web Console no longer encounters exceptions when IDERA Dashboard and the monitored instances are in different time zones.

SQLDM-30526 The Overview tab in the web console is no longer displaying an exception when clicking the summary under the Active Alerts widget.

Maintenance Mode

SQLDM-30686 SQL Diagnostic Manager no longer erroneously reporting "Maintenance mode started" when maintenance mode has been disabled.

Notifications

SQLDM-30859 The Databases category metrics now correctly raise alerts and send email notifications.

Query Monitor

SQLDM-30397 SQLDM now correctly creates historical snapshots when Query Monitor is enabled and configured to use Query Store.

SQLDM-30664 Powerscript now successfully enables Query Monitor.

Reports

SQLDM-21051 SQLDM successfully runs SSRS reports when more than 126 instances are selected.

Services

SQLDM-28648 SQLDM successfully collects information for disk drives volume formatted in ReFS.

Upgrade

SQLDM-30170 After upgrading SQLDM from 10.3 to 10.4 version, the Queries tab now correctly loads information.

SQLDM-30312 SQLDM now correctly triggers and notifies alerts for the Session CPU Time after performing an upgrade.

Known issues

IDERA strives to ensure our products provide quality solutions for your SQL Server needs. If you need further assistance with any issue, please contact [Customer Support Portal](#).

In this section you can find information on:

- [Issues known in this version.](#)
- [Issues known in recent releases.](#)
- [Issues known in releases previous to SQL Diagnostic Manager 10.0](#)

Known Issues in version 11.0

Reports

The 'Days since last backup metric' is based on the last time any kind of backup (Full, differential, or transaction log) was made on the database.

Azure Monitored Instances

The load time is longer for Azure screens because some features have been upgraded, such as:

- The 'Batch Collection' metric that has been replaced with individual collections makes the load time longer.
- The 'Initial Collections' in which the load time can be longer.

Reports

The Blocked Process Threshold is automatically updated when the activity monitor trace/extended event session starts, even if it already has the correct value.

By changing the Blocked Process Threshold, SQL Server reports that a configuration setting has changed. The "correct" value is set in the SQLDM instance properties under the Activity Monitor tab.

Sessions

The Session tab may not be visible if the database is unloaded and SQL Diagnostic Manager and System sessions are filtered.

Previous known issues

Known issues in version 10.6

Buttons

Refresh and Pause

The Refresh and Pause buttons are not currently working. Use the Function Keys temporarily, until this issue is fixed.

Reports

Deploy Report

The Deploy Report link under the Custom Report tab is temporarily unavailable, this link will be restored in SQLDM 11.0. The SQL Server Reporting Services can be used as a workaround.

Analyze tab

Overview

When you are in the Analyze tab you may not be able to return to the Overview tab. As a workaround, go to any other screen first and go back to Overview.

Known issues in version 10.5.1

There are no Known Issues in SQL Diagnostic Manager 10.5.1

Known Issues in version 10.5

Analysis


Workload Category


The Workload Analysis alerts categories are disabled by default to avoid inadvertent load. To run and ad hoc analysis, scroll down in the Run Analysis window and enable them.

Query Monitor

Estimated plan collection

On-demand estimated plan collection are not working for queries with bind variables.

 Customers on SQL Diagnostic Manager 10.4.2 and 10.4.3 should not attempt to upgrade to 10.5. If you have attempted to upgrade and encountered a failure, please contact [IDERA Technical Support](#).

 All the fixes included in 10.4.2 and 10.4.3 have not been integrated in SQL Diagnostic Manager 10.5. These Fix Versions are included in the upcoming version SQL Diagnostic Manager 10.5.1.

Known Issues in version 10.4

Query Store

Enabling

Query Store must be enabled on the monitored instance, along with Query Monitor and Query Waits.

IDERA Dashboard

Upgrading

Upgrading the IDERA Dashboard may fail if you have previously upgraded from version 3.8. Please contact [IDERA support](#) when encountering this problem.

Extended Events

Updates

When using Extended Events for Query Wait collections, updates happen at the snapshot interval. For real time updates, deselect Extended Event collection for the default of DMV sampling of Query Waits. Then, the graph will update with real-time collection intervals.

Query Details

Query Plan Diagram

When exporting a Query Plan Diagram you may encounter a Security Alert window with the following message "Your current security settings do not allow this file to be downloaded". Prevent this alert following these instructions:

1. Close SQLDM.
2. Open Internet Explorer.
3. In the menu bar, select **Internet Options**.
4. The Internet Options window opens, click the Security tab, select **Custom Level**, search for Downloads, and enable File Downloads.
5. Click **OK**.

Known Issues 10.3

Installation

Native client

SQL Diagnostic Manager installation may fail if Microsoft SQL Server Native Client is not installed on the machine. To avoid the failure, install the SQL Server native client first, and then run SQLDM installer. The native client installer can be found on the following page: [Microsoft® SQL Server® 2012 Native Client - QFE](#)

Privilege Minimization

User configuration

When configuring a privileged minimized user, the collection of WaitStatistics and QueryMonitoringStatistics could stop after SQLDM restarts.

IDERA Dashboard

Upgrade

In certain upgrades of the IDERA Dashboard, DM registration gets unregistered. If users attempt to register the product manually, it will fail because SQL Diagnostic Manager keeps unregistering on service restart.

Query Waits

Extended Events

When Query Waits is enabled with Extended Events selected, the CPU utilization will increase dramatically. To prevent this problem, deselect Extended Events for Query Wait collection.

Known Issues in version 10.2.3

IDERA Dashboard

Upgrade

When you upgrade SQL Diagnostic Manager, the installation wizard shows a message detecting the current version of your IDERA Dashboard. However, this message may also show that the Dashboard would be upgraded to an incorrect one.

Known issues in version 10.2.2

Licenses

License Library

The license library has changed. If you encounter problems with your current license, contact Licensing.

Installation and upgrade

Upgrading SQL Diagnostic Manager

When a local user that is not created as an administrator, but later is added to the local administrator group, runs the upgrade. It results in a restart through the upgrade and a partially completed upgrade.

Prevent this issue by running the upgrade under a user ID created or configured as a local administrator.

Known issues in version 10.2.1

IDERA Dashboard

Dashboard upgrade

After upgrading to SQL Diagnostic Manager 10.2.1 is necessary to stop and restart the SQLDM and the Dashboard services:

- Idera Dashboard Web Application Service
- Idera Dashboard Core Service
- SQLdm Rest Service
- SQLdm Predictive Analytics Service
- SQLdm Management Service
- SQLdm Collection Service

Follow the mentioned steps so that the Dashboard works as expected.

IDERA Dashboard

Dashboard Upgrade

IDERA Dashboard is not included in SQL Diagnostic Manager 10.2 upgrade. If you require IDERA Dashboard, you have three options:

1. Wait for SQLDM 10.2.1. A new release including a new IDERA Dashboard SQLDM interface is coming soon.
2. Upgrade SQLDM 10.1.4 with SQLDM 10.2 and leave the IDERA Dashboard in place. SQLDM 10.2 upgrade works with previous IDERA Dashboard integration, but new registrations are not supported.
3. Volunteer for the SQLDM 10.2.1 Beta. This release includes the IDERA Dashboard, but some known bugs may be present in the SQLDM integration with IDERA Dashboard.

Known issues in version 10.1.4

Installation

Failed installation on SQL Server 2005.

SQLDM is failing to install on SQL Server 2005 version. SQLDM 10.1.4 has encountered this issue when it is installed on SQL Server 2005 SP4.

WMI

WMI Service

WMI service must be running and be properly configured for disk space alerts to function normally.

Dashboard

Widgets

The instance status widget in the Dashboard is not loading any data.

Known issues in version 10.1.1

Installation

Failed SQLDM installation on Windows 10 or Windows 12.

Installation fails when the .Net 2.0/3.5 feature is not installed on Windows 10 or Windows 12. Therefore, install the .Net 2.0/3.5 feature before the SQLDM installation, once the installation is complete, the .Net 2.0/3.5 feature can be removed.

Known issues in version 10.1

PowerShell

Instance friendly name is not displayed in the UI.

The friendly name assigned to an instance through PowerShell is not being automatically refreshed and is not being displayed in the Desktop Console UI. Right-click **All Servers** in the tree pane and select **Refresh** to manually refresh the instances information and visualize the assigned friendly name.

IDERA Dashboard

Unsuccessful registration of SQLDM instances.

Installations of the IDERA Dashboard in 32 bit Operating Systems are unable to successfully register SQLDM instances hosted on 64 bit Operating Systems.

Known issues in version 10.0

License Manager

The Generate License Keys link fails to direct to License Manager.

SQL Diagnostic Manager version 10.0 is designed to integrate with License Manager 2.0, allowing for a self-service licensing model. However, an issue in the Generate License Keys link of the License Keys window fails to direct to License Manager. For additional information on this feature and a workaround, see [Generate a new license key](#).

IDERA Dashboard and web console

Views with list of instances available take longer to populate in the SQL Diagnostic Manager web console.

Some SQL Diagnostic Manager users may find that it takes approximately a minute for any of their monitored instances to display in views of the SQL Diagnostic Manager web console where a list of instances is available.

Known Issues in version 9.x

Known issues in version 9.1

General

The option to collect non-query activities data using Extended Events is incorrectly set by default for SQL Server 2008R2 instances.

SQL Diagnostic Manager is designed to support collection of non-query activities data with Extended Events only for instances running SQL Server 2012 or higher. In SQL Diagnostic Manager 9.1 for instances running SQL Server 2008R2 the Use Extended Events radio button is selected by default instead of the Use SQL Trace radio button.

Clicking the 'launch SQLdm' icon takes users to the corresponding instance dashboard screen but sometimes it fails to highlight the specific instance in the Servers tree.

SQL Diagnostic Manager is designed to allow users access to the SQL Diagnostic Manager Console from the Thumbnail, Heatmap, and Alerts Detail views. In SQL Diagnostic Manager 9.1 clicking the 'launch SQLdm' icon takes users to the corresponding instance dashboard screen but occasionally fails to clearly mark the specific instance it refers to in the Servers tree

Alerting

SQL Diagnostic Manager fails to display one alert response when multiple metric thresholds are breached

SQL Diagnostic Manager version 8.5 and later is designed to trigger an alert response only after multiple metric thresholds are breached when the "Where all metrics are in specified list" condition is selected. However, an issue in the "Send a Network Management (SNMP) Trap Message", "Send an email to recipient", and "Run the following SQL Script" actions of the Alert Response Window cause these actions to execute several times, resulting in multiple alerts.

The action "Run the following program" fails to execute when configuring automated responses to alerts

Users with SQL Diagnostic Manager version 8.0 and later may find that when creating an alert response with the associated action "Run the following program" in the Alert Response window, the specified program does not run once the condition is met.

SQL Diagnostic Manager returns PoweredOff alerts for the VM Power State metric when servers are not turned off

Some SQL Diagnostic Manager users may view PoweredOff alerts when reviewing the generated "VM Power State" alerts for their servers linked to vCenter or Hyper-V host servers. These alerts display even though the servers are not turned off.

The actions "Write an event to the Windows event log" and "Send an email to recipient" in the Alert Response Window fail to deliver multiple alert statuses

Users with SQL Diagnostic Manager version 8.5 and later may find that when creating an alert response with multiple metrics as a condition and associated actions "Write an event to the Windows event log" and/or "Send an email to recipient" in the Alert Response window, the feature fails to deliver complete alert information and displays only one alert status when condition is met.

VM Configuration

SQL Diagnostic Manager fails to update fields in the VM Configuration window after the virtualization host server is edited

Users with SQL Diagnostic Manager version 8.5 and later may find that fields such as the Virtual Machine name, Type, and Computer Name in the VM Configuration window are not updated correctly after the virtualization host server is changed.

Known issues in version 9.0

General

SQL Diagnostic Manager fails to update fields in the VM Configuration window after the virtualization host server is edited

Users with SQL Diagnostic Manager version 8.5 and later may find that fields such as the Virtual Machine name, Type, and Computer Name in the VM Configuration window are not updated correctly after the virtualization host server is changed.

Web Application Service

The IDERA Dashboard's Web Application Service fails to restart after reboot

Users with SQL Diagnostic Manager version 9.0 may find that the IDERA Dashboard's Web Application Service does not automatically restart after a machine reboot.

Alerting

SQL Diagnostic Manager fails to display one alert response when multiple metric thresholds are breached

SQL Diagnostic Manager version 8.5 and later is designed to trigger an alert response only after multiple metric thresholds are breached when the “Where all metrics are in specified list” condition is selected. However, an issue in the “Send a Network Management (SNMP) Trap Message”, “Send an email to recipient”, and “Run the following SQL Script” actions of the Alert Response Window cause these actions to execute several times, resulting in multiple alerts.

The action “Run the following program” fails to execute when configuring automated responses to alerts

Users with SQL Diagnostic Manager version 8.0 and later may find that when creating an alert response with the associated action “Run the following program” in the Alert Response window, the specified program does not run once the condition is met.

SQL Diagnostic Manager returns PoweredOff alerts for the VM Power State metric when servers are not turned off

Some SQL Diagnostic Manager users may view PoweredOff alerts when reviewing the generated “VM Power State” alerts for their servers linked to vCenter or Hyper-V host servers. These alerts display even though the servers are not turned off.

The actions “Write an event to the Windows event log” and “Send an email to recipient” in the Alert Response Window fail to deliver multiple alert statuses

Users with SQL Diagnostic Manager version 8.5 and later may find that when creating an alert response with multiple metrics as a condition and associated actions “Write an event to the Windows event log” and/or “Send an email to recipient” in the Alert Response window, the feature fails to deliver complete alert information and displays only one alert status when condition is met.

Known Issues in version 8.x

Known issues in version 8.6

Alerting

SQL Diagnostic Manager fails to update statistics for some monitored SQL servers and triggers a persistent alert 'Unable To Monitor' for the same

Users with SQL Diagnostic Manager version 8.0.1 and later may find that some servers are not returning any statistics due to an issue in the Activity Monitor settings that causes the entire refresh of servers to fail.

SQL Diagnostic Manager fails to display one alert response when multiple metric thresholds are breached

SQL Diagnostic Manager version 8.5 and later is designed to trigger an alert response only after multiple metric thresholds are breached when the "Where all metrics are in specified list" condition is selected. However, an issue in the "Send a Network Management (SNMP) Trap Message", "Send an email to recipient", and "Run the following SQL Script" actions of the Alert Response Window cause these actions to execute several times, resulting in multiple alerts.

The action "Run the following program" fails to execute when configuring automated responses to alerts

Users with SQL Diagnostic Manager version 8.0 and later may find that when creating an alert response with the associated action "Run the following program" in the Alert Response window, the specified program does not run once the condition is met.

The actions "Write an event to the Windows event log" and "Send an email to recipient" in the Alert Response Window fail to deliver multiple alert statuses

Users with SQL Diagnostic Manager version 8.5 and later may find that when creating an alert response with multiple metrics as a condition and associated actions "Write an event to the Windows event log" and/or "Send an email to recipient" in the Alert Response window, the feature fails to deliver complete alert information and displays only one alert status when condition is met.

SQL Diagnostic Manager returns PoweredOff alerts for the VM Power State metric when servers are not turned off

Some SQL Diagnostic Manager users may view PoweredOff alerts when reviewing the generated "VM Power State" alerts for their servers linked to vCenter or Hyper-V host servers. These alerts display even though the servers are not turned off.

SQL Diagnostic Manager fails to stop alert generation after the Availability Group Preferred Replica alert is unchecked

Users with SQL Diagnostic Manager version 8.6 that have the Availability Group Preferred Replica alert configured may find that when un-checking the alert in the Alert Configuration window to stop raising the alert, SQL Diagnostic Manager continues to generate the alert which also displays in the Alerts View. Un-checking the configured database in the Configuration tab of the Alert Configuration window solves this issue.

General

SQL Diagnostic Manager fails to update fields in the VM Configuration window after the virtualization host server is edited

Users with SQL Diagnostic Manager version 8.5 and later may find that fields such as the Virtual Machine name, Type, and Computer Name in the VM Configuration window are not updated correctly after the virtualization host server is changed.

Known issues in version 8.5

General

The Query Monitor trace in SQL Diagnostic Manager does not return queries in progress

Users with SQL Diagnostic Manager version 8.0 and later may find that when using the Query Monitor functionality, the queries returned and displayed do not include queries in progress.

SQL Diagnostic Manager fails to update fields in the VM Configuration window after the virtualization host server is edited

Users with SQL Diagnostic Manager version 8.5 may find that fields such as the Virtual Machine name, Type, and Computer Name in the VM Configuration window are not updated correctly after the virtualization host server is changed.

Alerting

SQL Diagnostic Manager returns Blocking Session Wait Time alerts with session 0 and missing information when blocking-process node is empty

Some users may experience Blocking Session Wait Time alerts with missing information and session 0 when the blocking-process node is empty in SQL Server. In SQL Diagnostic Manager 8.01 and later versions, this issue may occur only if the trace that is collecting blocking data doesn't return a SPID for a blocking process.

SQL Diagnostic Manager fails to display one alert response when multiple metric thresholds are breached.

SQL Diagnostic Manager version 8.5 and later is designed to trigger an alert response only after multiple metric thresholds are breached when the "Where all metrics are in specified list" condition is selected. However, an issue in the "Send a Network Management (SNMP) Trap Message", "Send an email to recipient", and "Run the following SQL Script" actions of the Alert Response Window cause these actions to execute several times, resulting in multiple alerts.

The action "Run the following program" fails to execute when configuring automated responses to alerts.

Users with SQL Diagnostic Manager version 8.0 and later may find that when creating an alert response with the associated action "Run the following program" in the Alert Response window, the specified program does not run once the condition is met.

The actions "Write an event to the Windows event log" and "Send an email to recipient" in the Alert Response Window fail to deliver multiple alert statuses.

Users with SQL Diagnostic Manager version 8.5 and later may find that when creating an alert response with multiple metrics as a condition and associated actions "Write an event to the Windows event log" and/or "Send an email to recipient" in the Alert Response window, the feature fails to deliver complete alert information and displays only one alert status when condition is met.

SQL Diagnostic Manager returns PoweredOff alerts for the VM Power State metric when servers are not turned off.

Some SQL Diagnostic Manager users may view PoweredOff alerts when reviewing the generated "VM Power State" alerts for their servers linked to vCenter or Hyper-V host servers. These alerts display even though the servers are not turned off.

Known issues in version 8.0

Installation and configuration issues

The SQLdm Setup Wizard is interrupted after selecting the SQL server instance on which you want to install the SQL Diagnostic Manager Repository

Some users may experience an interruption due to memory corruption in their SQL Diagnostic Manager installation after selecting the SQL Server instance on which they want to install their SQL Diagnostic Manager Repository. Note that the SQL Diagnostic Manager installer does not complete on machines where memory corruption prevents connections to SQL Server. Furthermore, in these types of environments other applications such as SQL Server Management Studio also do not function.

Negative values received from vCenter as a warning message do not impact VM data collection

Some users may receive negative values from vCenter as a warning message. The message is logged as in the SQL Diagnostic Manager Services. Note that this does not have any impact in terms of VM data collection and users are able to view VM and VM Host data.

Console crashes when upgrading to SQL Diagnostic Manager 8.0 while logged in the IDERA Newsfeed account

Users who remain logged in the IDERA Newsfeed account and upgrade to SQL Diagnostic Manager 8.0 may experience console crashes. Note that an upgrade from SQL Diagnostic Manager 8.0 to 8.0.1 will solve this issue.

Known Issues in version 7.x

Known issues in version 7.5

General

SQL Diagnostic Manager 7.5 no longer supports Microsoft Windows 2000 or the .Net 2.0 framework

SQL Diagnostic Manager 7.5 does not support Windows 2000 or the .Net 2.0 framework. While SQL Diagnostic Manager 7.2 and prior versions continue to operate with Windows 2000, SQL Diagnostic Manager 7.5 and later require the .Net 4.0 Full framework to take advantage of the additional features.

Server Overview

Timeline refresh causes a memory leak

SQL Diagnostic Manager experiences a memory leak when you are viewing the Timeline and click Refresh. To free the memory, you must restart the SQL Diagnostic Manager Management Console.

Queries

Queries view filter may not properly filter in some situations

In environments with large databases, the Queries view filter may not properly filter if you attempt to change filter properties during a refresh.

Resources

File Activity displays an unhelpful error message when WMI credentials are invalid

SQL Diagnostic Manager displays the error message, "An error occurred while refreshing the active view," when you attempt with access the Resources > File Activity view if your WMI credentials are invalid. If this message appears, verify your WMI access credentials and try again.

Installation and configuration considerations

SQL Diagnostic Manager Repository computer must run Microsoft SQL Server 2005 or later

SQL Diagnostic Manager 7.0 and later require Microsoft SQL Server 2005 or later running on the computer that hosts the SQL Diagnostic Manager Repository database for all installations.

SQL Diagnostic Manager Console only users cannot upgrade to the full version without uninstalling SQL Diagnostic Manager

Users who have an installation of SQL Diagnostic Manager Console only on a computer can use the upgrade to update their current version of the desktop client console, but cannot use the upgrade program to install the full version of SQL Diagnostic Manager.

Registering a SQL Server instance that uses an Alias Name requires an alias on the host computer

When registering a SQL Server instance using the alias name, an alias must first be created on the computer that hosts the SQLdm Management and Collection Services. If the instance uses Named Pipes, Named Pipes must be enabled in the SQL Server network configuration of the aliased SQL Server instance.

Management Service does not start when the account permissions are incorrect

If the account you specified for the SQL Diagnostic Manager services does not have System Administrator privileges on the SQL Server instance hosting the Repository database, the setup program cannot start the SQLdm Management Service. This issue only occurs during installation.

Some users may receive an error message when opening the SQL Diagnostic Manager stating that there is no valid license available

To resolve this issue, use the Management Service Configuration wizard to specify credentials for a Windows account or SQL Server login that has System Administrator privileges on the SQL Server instance hosting the Repository database. You can start the Management Service Configuration wizard from the Programs menu.

SQL Diagnostic Manager Reporting does not support Microsoft Reporting Services 2000

SQL Diagnostic Manager only supports Microsoft Reporting Services 2005 and later.

SQL Diagnostic Manager requires Predictive Analytics for baseline-related results

SQL Diagnostic Manager 7.5 requires the Predictive Analytics Service in order to display baseline-related results. During installations and upgrades, SQL Diagnostic Manager installs the Predictive Analytic service with the other included services. If Predictive Analytics is not detected or is disabled, SQL Diagnostic Manager cannot generate baseline data for the requested report. The following reports require the Predictive Analysis Service to include baseline-related results:

- [Baseline Statistics](#)
- [CPU Statistics](#)
- [Memory Statistics](#)
- [Session Statistics](#)

- [Disk Statistics](#)
- [Replication Statistics](#)

Phase out SQL Diagnostic Manager mobile manager

IDERA is announcing end of life for SQL mobile manager (SQLmm). On October 1, 2012, all support for SQLmm will stop and the SQLmm trial site will be decommissioned. Also, any remaining installations of SQLmm 3.6 or earlier that use the IDERA redirectors will no longer work.

To continue using a Web-based monitoring solution that is accessible from your smart phones or tablet devices, upgrade to SQL Diagnostic Manager 7.0 or later, and deploy SQLdm Mobile. IDERA discontinued distribution of SQL mobile manager in October 2011.

SQLdm Mobile does not support Microsoft IIS Express

SQLdm Mobile requires the standard version of the Microsoft IIS Web Server, version 7.0 or later. For more information, see the [SQLdm Mobile requirements](#).

SQLdm Mobile & Newsfeed installation cannot be changed or repaired using Add/Remove Programs

To successfully change or repair your SQLdm Mobile & Newsfeed installation, uninstall the application and components, and then run the setup program to reinstall your deployment.

SQLdm may not properly link to vCenter from some monitored virtual SQL Server instances

The SQLdm VM Configuration window displays a list of your monitored virtual SQL Server instances, but the link between an instance and the associated vCenter may not work properly. To access this window, click Tools > VM Configuration.

VM Host Server Change alert not always triggered when a VM moves from one host server to another

Even if the circumstances warrant SQL Diagnostic Manager to trigger the VM Host Server Change alert, users who have virtual machines running on an ESX host managed by vCenter 5.0 or later may not receive the alert when they move a VM from one host to another.

Negative values received from vCenter as a warning message do not impact VM data collection

Some users may receive negative values from vCenter as a warning message. The message is logged as in the SQL Diagnostic Manager Services. Note that this does not have any impact in terms of VM data collection and users are able to view VM and VM Host data.

Query Monitor disregards filter for SQL Server 2000 users

Some SQL Server 2000 users may notice that SQLdm Query Monitor includes SQL Diagnostic Manager queries in the results even if the Exclude SQL Diagnostic Manager queries check box is checked in the Advanced Query Monitor Configuration window.

SQLdm Mobile Server Dashboard charts display “Data not available” error

This error indicates that SQLdm Mobile was not able to collect all the performance metrics tracked by these charts. This issue is more likely to occur with the I/O and Per Read/Write charts on the Disk dashboard. To correct this issue, use the Microsoft IIS Manager console to restart the SQLdm Mobile Web site (virtual directory).

Start and Stop service actions not supported for virtual SQL Server instances

The Start and Stop service actions are not supported for virtual SQL Server instances (instances located on a Windows cluster node). To start or stop a service on a virtual SQL Server 2008 instance, use the Microsoft

Failover Cluster Management tool. For SQL Server 2005 or earlier, use the Microsoft Cluster Administrator tool to manage services.

Gaps between the real-time view and back-filled historical data can be seen on the CPU view

This occurs only when the CPU view is accessed via the Server tree. To view the information, without any gaps, access the CPU view via the ribbon.

Free disk space reported as zero on databases with special characters

For servers and path names that include special characters such as % or #, SQL Diagnostic Manager may incorrectly report the remaining disk space as zero.

Known Issues in versions previous to 7.5

Known issues in versions previous to 7.5

General

SQL Diagnostic Manager displays a refresh error when a user attempts to view data for an instance to which he does not have admin rights

SQL Diagnostic Manager displays a refresh error message when a user tries to view data for a monitored SQL Server instance to which the user does not have administrative rights. The message states that, "An error occurred while refreshing the active view."

SQL Diagnostic Manager displays an error when a user attempts to configure disk collection for an instance to which he does not have admin rights

SQL Diagnostic Manager displays an error message on the Server Properties Disk Drives window if a user's collection service credentials do not have administrative access to the monitored SQL Server instance, and the user attempts to view the list of attached disk drives on that instance. Note that WMI credentials are not used to retrieve disk drive information on this page.

Drives may not appear in the Resources Disk view when you switch to direct WMI collection

Users who select to collect OS metrics using direct WMI collection may notice mounted drives appearing in some areas of SQL Diagnostic Manager but not on the Resources Disk view. To refresh this list, you must restart the SQL Diagnostic Manager Management Console.

Dashboard

Custom Counters dashboard panel does not return to default after deleting all counters

The Custom Counters dashboard panel includes a drop-down list to allow you to select from the available counters. If you delete all counters, the panel does not return to the default title and continues to display one of the counter names.

Resources

Disk Filter Columns does not work on a blank chart

The Filter Columns selection does not work in the Resources > Disk view if you receive a blank chart due to your WMI credentials being invalid.

Alerts

OS Disk Time (Percent) and OS Disk Time Per Disk (Percent) alerts may not trigger for some users

Users with Windows Server 2008 SP1 and SQL Server 2000 SP4, and who select direct WMI collection for their OS metrics may experience an issue preventing OS Disk Time (Percent) and OS Disk Time Per Disk (Percent) alerts from triggering. If this issue occurs, switch to using OLE automation.

Virtualization-related alerts occasionally raise more than once

In some circumstances, SQL Diagnostic Manager may repeatedly raise virtualization-related alerts when there is no change in the status.

Custom Counters

SQL Diagnostic Manager does not automatically add unassociated custom counters to an alert template created from an instance

If you use alert templates and create a custom counter, SQL Diagnostic Manager automatically adds the new counter to your existing alert templates. However, if you create a custom counter not associated with an instance, and then create an alert template based on the alert configuration of one of your monitored instances, SQL Diagnostic Manager does not add the new counter to your alert template nor can you manually add the counter.

To avoid this issue, add the missing custom counter to any SQL Server instance, create an alert template from that instance, and then clear any undesired custom counters from that instance.

SQLdm Mobile

SQLdm Mobile My Servers view filter does not work

The filter on this view will consistently display the message, "You have no favored or followed servers or the servers list has not been cached yet," regardless of the filter option you choose. This issue occurs even when you are following servers or have marked servers as favorites.

Recommended IDERA Solutions

IDERA strives to ensure our products provide quality solutions for your database needs. The following IDERA Solutions have been recently added to the knowledge base at the [Customer Service Portal](#).

ID Number	Title
4337	SQLDM alert, "Session 0 has been blocking other sessions..."
4287	Disk Full Percent fires at or near 100% for a disk that still has plenty of space.
4281	A SQL Diagnostic Manager with wait type MSQL XP uses constant CPU resources and never finishes on a monitored server.
4088	Disk Busy per Disk counter reports 100% when the disk is known to be idle.
3511	SQLDM always shows "The SQLDM Collection Service has too many outstanding connections" in the overview pane for a specific server, and SPIDs stall on the audited server.
2286	When Upgrading SQL Diagnostic Manager, you receive the error message "The repository is an invalid version."
2109	CPU Used, Memory Paging and Disk Queue Length show question marks in the Summary view.
2102	What alerting Event Codes are used in SQLDM 5.0 or later?

Get started

SQL Diagnostic Manager (SQLDM) is a database management system (DBMS) solution designed exclusively for Microsoft SQL Server diagnosis and performance tuning. SQL Diagnostic Manager lets you discover potential and existing problems, analyze the state of your SQL Server enterprise, and generate reports.

Upgrade to this release

Upgrading SQL Diagnostic Manager to the most recent version allows you to take advantage of the new features available. An up-to-date list of new features and fixed issues is available in the [SQL Diagnostic Manager Release Notes](#).

Upgrade from SQL Diagnostic Manager 5.x or higher to SQL Diagnostic Manager 10.6 checklist

To successfully upgrade your Microsoft SQL Server environment to this build, complete the steps included in the following table:

✓	Follow these steps ...
	Back up your SQL Diagnostic Manager Repository prior to upgrading.
	Make sure that your environment meets the product requirements for the newer version of SQL Diagnostic Manager .
	Verify that no SQL Diagnostic Manager desktop clients are running.
	Upgrade your environment to the latest version of SQL Diagnostic Manager on the computer where the SQL Diagnostic Manager services reside. Note that SQL Diagnostic Manager 10.0 and later versions integrate with the IDERA Dashboard.
	Test your new SQL Diagnostic Manager deployment and setup .
	Upgrade your IDERA Newsfeed and SQLdm Mobile installations or deploy a new IDERA Newsfeed and SQLdm Mobile installation .

Upgrade to SQLDM 9.0 or higher versions

Customers upgrading from previous SQL Diagnostic Manager versions to SQL Diagnostic Manager 9.0 or higher versions, need a new license key or it is not possible to continue with the upgrade process. Users can access the IDERA Customer Portal at the moment of upgrade from the license management window in SQL Diagnostic Manager (**Help > Manage Licenses**) or directly at <https://idera.secure.force.com>. Information necessary to generate a new license key is provided in the license management screen or the portal.

Upgrade SQL Diagnostic Manager in a clustered environment

Use the following steps to upgrade your SQL Diagnostic Manager installation in a clustered environment.

- ✔ SQL Diagnostic Manager 10.0 and later requires Microsoft SQL Server 2005 running on the computer that hosts the SQL Diagnostic Manager Repository database for all installations.

Upgrade in a Microsoft Windows Server 2003 Clustered Environment

To upgrade SQL Diagnostic Manager in a clustered environment:

1. Log on to the cluster node using the Administrator account.
2. In Cluster Administrator, select the resource group that contains the SQL Diagnostic Manager services.
3. Take the SQLDM Management and Collection services off line.

Perform the following steps on each node, starting with the active node:

1. [Upgrade SQL Diagnostic Manager components previously deployed to the node.](#)
2. In Windows Services, stop the SQLDM Collection, SQLDM Management, and SQLDM Predictive Analytics services.
3. Set the SQLDM services to manually start, and then restart the services.

Once the previous steps are complete, perform the following steps on each cluster node:

1. In Cluster Administrator, bring the SQLDM Management Service on line.
2. Use the Management Service Configuration wizard to force the services to re-register on the active node.

Upgrade in a Microsoft Windows Server 2008 Clustered Environment

To upgrade SQL Diagnostic Manager in a clustered environment:

1. Log on to the cluster node using the Administrator account.
2. In the Failover Cluster Management tool, select the resource group that contains the SQL Diagnostic Manager services.
3. Take the SQLDM Management and Collection services offline.

Perform the following steps on each node, starting with the active node:

1. [Upgrade SQL Diagnostic Manager components previously deployed to the node.](#)
2. In Windows Services, stop the SQLDM Collection, SQLDM Management, and SQLDM Predictive Analytics services.
3. Set the SQL Diagnostic Manager services to manually start, and then restart the services.

Once the previous steps are complete, perform the following steps on each cluster node:

1. Bring the SQLDM Management Service on line.
2. Use the Management Service Configuration wizard to force the services to re-register on the active node.

Perform a standard upgrade

Use the following procedures to upgrade a distributed SQL Diagnostic Manager environment.

Upgrade your SQLDM Repository and/or SQLDM Services

Install the SQLDM Repository database on a SQL Server instance that [meets the product requirements](#). You do not need to install your Repository in the same domain as the SQLDM Console, but a two-way trust between the target domains must exist. You can install the SQLDM services on the Repository computer (recommended) or a different computer.

- ✔ SQL Diagnostic Manager 10.1 and later requires Microsoft SQL Server 2005 running on the computer that hosts the SQLDM Repository database for all installations.

To upgrade the SQLDM Repository and/or Services:

1. Use an administrator account to log on to the computer on which you want to upgrade the SQLDM Repository database and/or Services.
2. Run Setup.exe in the root of the installation kit.
3. On the IDERA SQL Diagnostic Manager Quick Start window, click **SQL Diagnostic Manager** under the Install heading.
4. Read the Upgrade Wizard Welcome window, and then click **Next**.
5. Upgrade your Repository by verifying the authentication method and connection credentials, and then click **Next**.
6. Upgrade your Service account by specifying the account credentials that you want the SQLDM services to use to connect to the SQLDM Repository and the monitored SQL Server instances, and then click **Next**. This account must belong to the sysadmin role on the Repository database and each instance you plan to monitor.
7. Click **Upgrade** to begin your upgrade.
8. Click **Finish** to exit the upgrade program.

Upgrade your SQLDM Console

- ✔ You must have a full installation of SQL Diagnostic Manager already running in your environment before attempting to upgrade the SQLDM desktop client console.

To upgrade the SQLDM Console:

1. Use an administrator account to log on to the computer on which you want to upgrade the SQLDM Console.
2. Navigate to the location where the SQLDM Console file resides in the distribution media.
3. **If you use an x64 operating system**, run SQLDiagnosticManager-x64_c.exe. **If you use an x86 OS**, run SQLDiagnosticManager_c.exe.
4. On the Welcome window of the setup program, click **Next**.
5. Click **Upgrade** to begin your upgrade.
6. Click **Finish** to exit the upgrade program.

Upgrade SQLDM console only

SQL Diagnostic Manager allows users to install a full version, which includes the SQL Diagnostic Manager Repository, Management Service, and Management Console, or you can install the SQLDM desktop client console by itself when you already have a full version of SQL Diagnostic Manager installed in your environment.


Users who have an installation of SQLDM Console only on a computer can use the upgrade to update their current version of the desktop client console, but cannot use the upgrade program to install the full version of SQL Diagnostic Manager.

Test your deployment

To test your upgrade, you can immediately begin collecting and alerting on performance data. Verify that you are able to monitor the target SQL Server instances and that the data you are retrieving is valid.

Installation and deployment

You can install SQL Diagnostic Manager and SQLDM Mobile in an environment of any size.

 To install the Trial for SQL Diagnostic Manager, review the guide in the [Trial installation instructions](#).

Get ready for your SQL Diagnostic Manager and IDERA Dashboard deployment

- Check the [supported installation scenarios](#)
- Learn about the [components and architecture](#)
- Review [system requirements](#)
- View the [installation instructions](#)

Get ready for your SQLDM Mobile or IDERA Newsfeed deployment

- Check the [supported installation scenarios](#)
- Learn about the [components and architecture](#)
- Review [system requirements](#)
- View the [installation instructions](#)

Get ready for your SQL Diagnostic Manager Console only deployment

- Check the [supported installation scenarios](#)
- Learn about the [components and architecture](#)
- Review [system requirements](#)
- View the [installation instructions](#)

Supported installation scenarios

You can install and deploy SQL Diagnostic Manager, SQLDM Mobile, and the IDERA Newsfeed to meet your unique SQL Server environment needs. SQL Diagnostic Manager also offers a SQL Diagnostic Manager Console-only installation in environments that already have SQL Diagnostic Manager fully installed. For additional information about this type of installation, see [deploy SQL Diagnostic Manager Console only](#).

The following installation scenario descriptions assume you are deploying SQL Diagnostic Manager, SQLDM Mobile, and IDERA Newsfeed in your environment.

✓ In larger environments that contain more than one SQL Diagnostic Manager installation, consider dedicating a SQLDM Mobile deployment per each SQL Diagnostic Manager application.

✓ SQL Diagnostic Manager supports monitoring virtual SQL Server instances, such as those using VMware or other virtualization software. For additional information about how SQL Diagnostic Manager works with virtualization, see [How SQL Diagnostic Manager works with a virtual environment](#).

i SQL Diagnostic Manager supports running **from** Amazon EC2 or Azure IaaS environments.

Typical environment

This configuration includes the following installations:

- SQL Diagnostic Manager Console on your workstation
- SQLDM Mobile Web site on a Web Server for smartphone access
- SQLDM Mobile and Newsfeed Service on the same computer that hosts the SQL Diagnostic Manager services
- SQLDM Mobile Repository and SQL Diagnostic Manager Repository on the same SQL Server instance

Clustered environment

You can install and configure SQL Diagnostic Manager to monitor virtual SQL Servers. A virtual SQL Server is a SQL Server running on a Microsoft failover cluster managed by Microsoft Cluster Services.

For SQL Diagnostic Manager to monitor virtual SQL Servers, you need to point SQLDM at the "Virtual" SQL Server rather than the individual members of the cluster.

You can install and configure SQL Diagnostic Manager with its core services and repository databases on a Microsoft failover cluster managed by Microsoft Cluster Services. For more information, review [Deploy SQL Diagnostic Manager in a clustered environment](#).

Web farm environment

You can install SQLDM Mobile in a Web farm to make sure this site remains accessible during a failover situation. This configuration includes the following installations:

- SQLDM Mobile on a Web Server in the farm
- Typical SQL Diagnostic Manager installation
- SQLDM Mobile and Newsfeed Service on the same computer that hosts the SQL Diagnostic Manager services
- SQLDM Mobile Repository and SQL Diagnostic Manager Repository on the same SQL Server instance

Non-trusted environment

You can install and configure SQL Diagnostic Manager and SQLDM Mobile to monitor SQL Server instances running in non-trusted domains or workgroups.

This configuration includes the following installations:

- SQL Diagnostic Manager Console on your workstation in a trusted or non-trusted domain
- SQLDM Mobile Web site on a Web Server for smartphone access
- SQLDM Mobile & Newsfeed Service on the same computer that hosts the SQL Diagnostic Manager services, in a trusted or non-trusted domain
- SQLDM Mobile Repository and SQL Diagnostic Manager Repository on the same SQL Server instance, in a trusted or non-trusted domain

Choose your deployment strategy

Before you begin installing SQL Diagnostic Manager, it is important to consider your deployment choices.

- ✔ You can deploy SQLDM Mobile after you have installed SQL Diagnostic Manager. For more information, review the SQLDM Mobile [architecture and components](#), as well as [requirements](#).

Trial deployment

A trial deployment installs each SQL Diagnostic Manager component on a separate, dedicated computer. The IDERA evaluation license is fully functional and monitors up to 15 SQL Server instances anywhere on your network for a period of 15 days. For more information, review the [Trial installation instructions](#).

Production deployment

A production deployment installs each SQL Diagnostic Manager component on a separate, dedicated computer. This deployment supports stable performance and availability as your environment grows and your monitoring needs change.

Deploy the IDERA Dashboard and SQL Diagnostic Manager

Use the following links to prepare for your IDERA Dashboard and SQL Diagnostic Manager deployment:

- Check the [supported installation scenarios](#)
- Learn about the [components and architecture](#)
- Review [system requirements](#) for the IDERA Dashboard and SQL Diagnostic Manager
- View the [installation instructions](#)
- [Log in](#) the IDERA Dashboard and SQL Diagnostic Manager web console

Product components and architecture

The IDERA Dashboard and SQL Diagnostic Manager consist of a light, unobtrusive architecture that easily runs in your SQL Server environment with minimal configuration. All components run outside and separate from SQL Server processes.

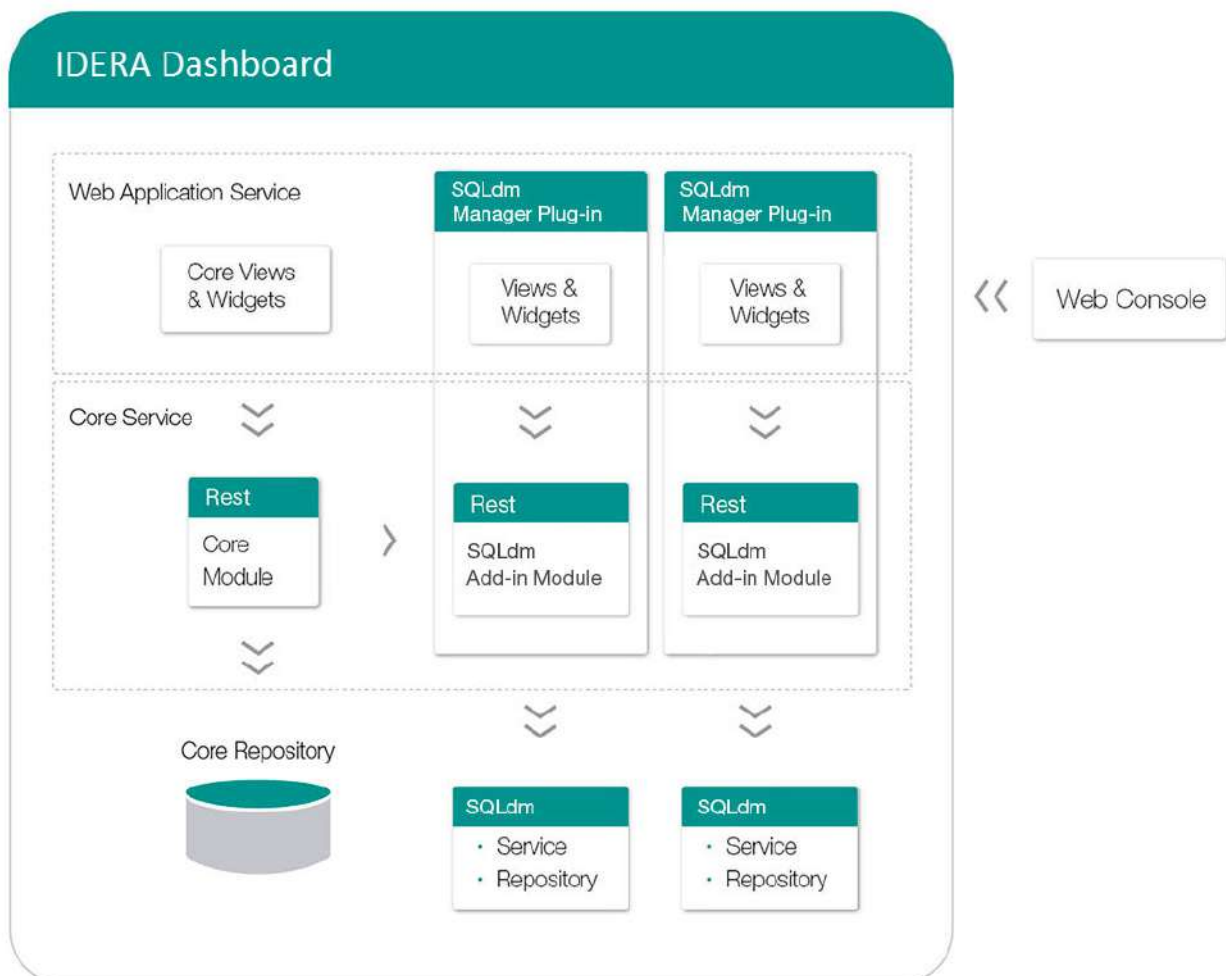
- Learn about the components and architecture of the [IDERA Dashboard](#).
- Learn about the components and architecture of the [SQL Diagnostic Manager](#).

IDERA Dashboard components and architecture

The IDERA Dashboard provides web and back end services, shared across multiple IDERA products. To learn more about what the IDERA Dashboard is and how it works, see [Navigate the IDERA Dashboard](#).

The IDERA Dashboard consists of the following components:

- [Web Application Service](#)
- [Core Service](#)
- [Core Repository](#)



Web Application Service

The Web Application Service is a Windows service that wraps Apache Tomcat server. The Web Application Service serves up dashboard (IDERA Dashboard) and SQLdm views and widgets that are displayed in the web console. The Web Application Service requires three ports:

- Standard HTTP port (by default 9290)
- Monitor port (9094)
- SSL port (9291)

Core Service

The Core Service is a C# (.NET 4.0 Framework) based Windows service that hosts dashboard and SQLDM REST APIs that are used by the Web Application Service to configure and retrieve data. In addition, the Core Service handles product registration, security, configuration, product data, and event aggregation.

The Core Service uses two ports, one for REST API and the other for .NET remoting:

- Core Service REST API port (by default 9292)
- .NET remoting port (by default 9293)

Core Repository

The Core Repository is a database where all dashboard configuration and aggregated data is stored. The Core Repository database is hosted on a SQL Server instance and is accessed by the Core Service to retrieve data.

SQL Diagnostic Manager components and architecture

SQL Diagnostic Manager consists of a light, unobtrusive architecture that easily runs in your SQL Server environment with minimal configuration. All SQL Diagnostic Manager components run outside and separate from SQL Server processes.

SQL Diagnostic Manager Console

Use the SQL Diagnostic Manager Console to:

- View real-time status
- Configure alert notifications on specific metric thresholds at the server and database levels
- View historical reports
- Perform administrative functions

The SQL Diagnostic Manager Console retrieves historical information directly from the SQL Diagnostic Manager Repository. All real-time requests use the SQL Diagnostic Manager services to poll the monitored SQL Server.

SQL Diagnostic Manager Services

SQL Diagnostic Manager has four centralized services, the Management Service, the Collection Service, the Predictive Service, and Rest Service. These four services reside on the same computer.

The Management Service

The Management Service performs the following primary functions:

- Provides real-time data to the SQL Diagnostic Manager Console
- Receives historical data from the Collection Service for storage in the Repository
- Raises alerts and sends alert notifications

The Collection Service

The Collection Service performs on-demand and scheduled collection from the monitored SQL Servers.

The Predictive Service

The Predictive Service is used for Alert Forecasting and performs the following primary functions:

- Calculates the Alert Forecast every hour
- Builds a forecasting model once a day

The Rest Service

SQL Diagnostic Manager Plug-in

When SQL Diagnostic Manager is registered with the IDERA Dashboard, the product plug-in module is deployed. The SQL Diagnostic Manager plug-in consists of web views and widgets and a .NET based add-in module (SQL Diagnostic Manager add-in). The web views and widgets are deployed in the Web Application Service of the IDERA Dashboard, and the SQL Diagnostic Manager add-in in the Core Service of the same.

The Web Application Service dynamically loads in SQL Diagnostic Manager's views and widgets and makes them available to web console users. The views and widgets use the SQL Diagnostic Manager add-in REST APIs to retrieve data. Likewise, the SQL Diagnostic Manager add-in retrieves data from the product services and Repository.

SQL Diagnostic Manager Repository

The SQL Diagnostic Manager Repository is a centralized SQL Server database that stores collected metrics on a scheduled basis, historical data, and alerts information. The SQL Diagnostic Manager Repository also stores configuration information, such as the credentials used to monitor a registered SQL Server instance.

- ✔ SQL Diagnostic Manager 10.1 and later requires Microsoft SQL Server 2005 or above running on the computer that hosts the SQL Diagnostic Manager Repository database for all installations.

Authentication in SQL Diagnostic Manager

SQL Diagnostic Manager uses the same types of authentication available in the SQL Server security model. When specifying account credentials for the SQL Diagnostic Manager services, you can use Windows Authentication or SQL Server Authentication.

When considering which authentication to use, keep in mind that SQL Server Authentication is required when no domain trust exists between the SQL Diagnostic Manager Services computer and the computers hosting the monitored SQL Server instances. For example, if the monitored SQL Server instances are located in an untrusted domain or behind a firewall, you must use SQL Server Authentication to successfully deploy SQL Diagnostic Manager. In this case, you must use the **sa** account or a SQL Server login that has System Administrator permissions.

For more information about these authentication types and the SQL Server security model, see the Microsoft document, [Authentication in SQL Server](#).


Product requirements

The IDERA Dashboard and SQL Diagnostic Manager consist of a light, unobtrusive architecture that easily runs in your SQL Server environment with minimal configuration. Review system requirements for the IDERA Dashboard and SQL Diagnostic Manager.

- Learn about the [IDERA Dashboard requirements](#).
- Learn about the [SQL Diagnostic Manager requirements](#).

IDERA Dashboard and web console requirements


To successfully install the IDERA Dashboard and SQL Diagnostic Manager web console you need to comply with the requirements described in the [Software Requirements](#) page for the IDERA Dashboard.


 The IDERA Dashboard Web Application service comes with SSL already set up. For more information on running the IDERA Dashboard over SSL, see [Run the Idera Dashboard over SSL \(HTTPS\)](#).

SQL Diagnostic Manager requirements

There are three types of SQL Diagnostic Manager installation (Trial, General, and Production). Refer to the following list of requirements for each type of installation.

 SQL Diagnostic Manager 10.5 and above requires Microsoft SQL Server 2005 or above running on the computer that hosts the SQL Diagnostic Manager Repository database for all installations.

 Tempdb monitoring is supported on SQL Server 2005 and above servers only.

 To run PowerShell scripts, upgrade to PowerShell 3.0 or higher. This upgrade is required to use the PowerShell snap-in.

Trial requirements

Consider the following requirements when installing all SQL Diagnostic Manager components on one computer.

Type	Requirement
SQL Server Standard or Enterprise Edition	<ul style="list-style-type: none"> • SQL Server 2005 SP1 or later • SQL Server 2008 or later • SQL Server 2008 R2 • SQL Server 2012 • SQL Server 2014 • SQL Server 2016 • SQL Server 2017 • SQL Server 2019
Hard Disk Space	200 MB for all SQL Diagnostic Manager components
Windows Permissions	Your Windows Logon account includes local Administrator permissions
SQL Server Privileges	Either a Windows user account or SQL Server login that includes System Administrator privileges on the monitored SQL Server instances

General requirements

The following requirements apply for all SQL Diagnostic Manager components.

Type	Requirement
CPU	2.0 GHz or higher

Type	Requirement
Memory	1 GB
Operating System 64-bit	<ul style="list-style-type: none"> • Windows Server 2008 SP1+ • Windows Server 2008 R2 • Windows 7 • Windows 8 • Windows Server 2012 • Windows Server 2012 R2 • Windows 10 • Windows Server 2016 • Windows 2019
Port Access	5166, 5167, 5169
Client Browser	Internet Explorer 7.0 or later

Production requirements

Consider the following requirements when installing SQL Diagnostic Manager in an environment where the components are on separate computers.


SQL Diagnostic Manager Console

Type	Requirement
Disk Space	40 MB
SQL Server Permissions	Your Windows login account includes Read Access to the SQL Diagnostic Manager Repository. You can specify a SQL Server login instead.
Monitor Resolution	1280 x 1024 with small text for the best viewing of the SQL Diagnostic Manager Dashboard


SQL Diagnostic Manager Services

Type	Requirement
Disk Space	50 MB
Collection Service Permissions	For more information, review the User Creation Scripts .
Management Service Permissions	Sysadmin fixed server role on SQL Diagnostic Manager Repository instance

Type	Requirement
Linux - SQL Server Server 2017 (Monitored SQL Instances only)	<ul style="list-style-type: none"> • Red hat Enterprise 7.3 or Higher • SUSE Enterprise v12 SP2 or Higher • Ubuntu 16.04LTS and 16.10 or Higher
Windows Runtime Components	<ul style="list-style-type: none"> • Microsoft .NET 4.7.2+ (Required) • MDAC 2.8

 SQL Diagnostic Manager 11.0 installer checks for your current .NET framework version and prompts the .NET 4.7.2 update. After installing the latest version of .NET Framework you need to reboot your machine; otherwise, SQLDM installation fails.

SQL Diagnostic Manager Repository;

 SQL Diagnostic Manager 7.2 and later includes a repository autogrow setting at 32 MB, which is sufficient for trial and small environments in production. Environments with larger, multi-server installations should change the repository autogrow setting to 250 MB or higher to prevent frequent data file autogrow, which can cause performance problems. Using percentage-based autogrow is not recommended.

Type	Requirement
Disk Space	1GB to 3GB for one year of data per monitored SQL Server instance
SQL Server Express, Standard, or Enterprise/ Datacenter Edition	<ul style="list-style-type: none"> • SQL Server 2008 or later • SQL Server 2008 R2 • SQL Server 2012 • SQL Server 2014 • SQL Server 2016 • SQL Server 2017 (Windows & Linux) • SQL Server 2019
Cloud Deployment	<ul style="list-style-type: none"> • Azure IaaS VM Linux (SQL Server 2019) • Amazon AWS IaaS VM Windows • Amazon AWS IaaS Linux (SQL Server 2019)

Monitored SQL Server Instances

Type	Requirement
SQL Server Express, Standard, or Enterprise/ Datacenter Edition	<ul style="list-style-type: none"> • SQL Server 2005 • SQL Server 2008 R2 • SQL Server 2012 • SQL Server 2014 • SQL Server 2016 • SQL Server 2017 (Windows & Linux) • SQL Server 2019 (Windows & Linux) • SQL Azure • Amazon RDS
Cloud Deployment	<ul style="list-style-type: none"> • Azure DB: Single, Elastic Pool, Managed Instance. • Amazon RDS SQL: SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, SQL Server 2016, SQL Server 2017, SQL Server 2019 • Azure IaaS VM Windows (All SQL Versions) • Azure IaaS VM Linux (SQL Server 2017, 2019) • Amazon AWS IaaS VM Windows (All SQL Versions) • Amazon AWS IaaS VM Linux (SQL Server 2017)

📘 Azure SQL and Amazon RDS permissions

Review the [Minimum permissions to monitor an Amazon RDS instance](#) and the [Minimum and Recommended Permissions for Azure SQL](#) to create a SQL Server user on the server to monitor a cloud instance.

This topic provides the minimum requirements for installing SQL Diagnostic Manager. For more information about installation requirements, view [SQL Diagnostic Manager Hardware Sizing guidelines](#).

SQL Diagnostic Manager Management Pack and SCOM support requirements

The IDERA SQL Diagnostic Manager Management Pack for integration with SCOM is designed for System Center Operations Manager 2007 R2 and above. For more information about SQL Diagnostic Manager MP and SCOM installation, see [Install and configure the SQL Diagnostic Manager Management Pack for SCOM](#).

Minimum permissions to monitor an Amazon RDS instance

To monitor an Amazon RDS instance using SQL Diagnostic Manager, you have to create a SQL Server user on the instance with the following minimum permissions:

Server Roles:

public, processadmin, setupadmin

Databases:

public on TempDB

sysadmin server role equivalent privileges for monitored databases

Securables:

Server Object:

Alter any connection

Alter any linked server

Alter any login

Alter any server role

Alter server state

Alter trace

Connect SQL

Create any database

Create server role

View any database

View any definition

View server state


SQL Diagnostic Manager Hardware Sizing guidelines

The following guidelines provide an estimation of the hardware resources required to deploy SQL Diagnostic Manager depending on the number of servers you want to monitor with SQLDM.

Installs under 20 servers

Single-server deployment - Includes the SQL Diagnostic Manager Services and the SQL Server hosting the SQLdm Repository (clustered for redundancy).


Type	Requirement
Operating System	Windows Server 2012 Windows Server 2016 Windows Server 2019
Memory	8 GB
CPU	6 Core
SQLdm Repository size	Approximately 50GB
SQL Server version	SQL Server 2008R2 standard edition or higher

 The expected repository growth is approximately 1.5-3 GB per monitored instance for a year worth of collected data. Monitoring 15 instances could gather approximately 45 GB over the course of a year.

Installs 20-50 servers

Single-server deployment - Includes the SQL Diagnostic Manager Services and the SQL Server hosting the SQLdm Repository (virtualized/clustered for redundancy).

Type	Requirement
Operating System	Windows Server 2012 Windows Server 2016 Windows Server 2019
Memory	16 GB
CPU	6 Core
SQLdm Repository size	Approximately 150GB
SQL Server version	SQL Server 2008R2 standard edition or higher

 The expected repository growth is approximately 1.5-3 GB per monitored instance for a year worth of collected data. Monitoring 30 instances could gather approximately 90 GB over the course of a year.

Installs 51-200 servers

Dual-server deployment - First server hosts the SQL Diagnostic Manager Services and second server hosts the SQL Server and the SQLdm Repository database.

- **Machine 1** - Hosting the SQL Diagnostic Manager services (virtualized/clustered for redundancy)

Type	Requirement
Operating System	Windows Server 2012 Windows Server 2016 Windows Server 2019
Memory	8-16 GB
CPU	6 Core

- **Machine 2** - Hosting the SQL Server and the SQLdm Repository (clustered for redundancy)

Type	Requirement
Operating System	Windows Server 2012 Windows Server 2016 Windows Server 2019
Memory	16 GB
CPU	6 Core
SQLdm Repository size	200-600GB
SQL Server version	SQL Server 2008R2 standard edition or higher

Installs 201-300 servers

Dual-server deployment - First server hosts the SQL Diagnostic Manager Services and second server hosts the SQL Server and the SQLdm Repository database.

- **Machine 1** - Hosting the SQL Diagnostic Manager services (virtualized/clustered for redundancy)

Type	Requirement
Operating System	Windows Server 2012 Windows Server 2016 Windows Server 2019
Memory	16 GB
CPU	8 Core

- **Machine 2** - Hosting the SQL Server and the SQLdm Repository (clustered for redundancy)

Type	Requirement
Operating System	Windows Server 2012 Windows Server 2016 Windows Server 2019
Memory	32 GB
CPU	8 Core
SQLdm Repository size	600 GB or more
SQL Server version	SQL Server 2008R2 standard edition or higher

Installs over 300 servers

Dual-server deployment - First server hosts the SQL Diagnostic Manager Services and second server hosts the SQL Server and the SQLdm Repository database.

- **Machine 1** - Hosting the SQL Diagnostic Manager services (virtualized/clustered for redundancy)

Type	Requirement
Operating System	Windows Server 2012 Windows Server 2016 Windows Server 2019
Memory	32 GB
CPU	8 Core

- **Machine 2** - Hosting the SQL Server and the SQLdm Repository (virtualized/clustered for redundancy)

Type	Requirement
Operating System	Windows Server 2012 Windows Server 2016 Windows Server 2019
Memory	32 GB
CPU	8 Core
SQLdm Repository size	900 GB or more
SQL Server version	SQL Server 2008R2 standard/enterprise edition

Sample installations in large environments

See the following sample customer installations for users with large environments:

Installs 51-100 servers

Type	In-House Use
Number of monitored servers	52
Services and Repository installed on same machine	Hyper-V <ul style="list-style-type: none"> • 4 quad core processors • 14 GB of memory • Windows Server 2008 R2 Enterprise
Data storage and grooming options	<ul style="list-style-type: none"> • Repository size: 120 GB • Retain 60 days of history
Monitoring features	Monitor Windows and SQL Server performance and error log monitoring

Type	In-House Use
Number of monitored servers	90
Services machine	<ul style="list-style-type: none"> • 2 vCPU • 8 GB of memory • Windows Server 2008 R2 Enterprise
Services machine	<ul style="list-style-type: none"> • 2 vCPU • 8 GB of memory • Windows 2008 R2 Enterprise • Repository: SQL Server 2008 R2
Data storage and grooming options	<ul style="list-style-type: none"> • Repository size: 100 GB • Retain 60 days of history
Monitoring features	Monitor Windows and SQL Server performance

Installs 201- 300 servers

Type	In-House Use
Number of monitored servers	234
Services and Repository installed on same machine	<ul style="list-style-type: none"> • Dell 2950 • 2 quad core processors • 16 GB of memory • Windows Server 2003
Data storage and grooming options	<ul style="list-style-type: none"> • Repository size: 140 GB • Retain 30 days of history

Type	In-House Use
Monitoring features	Monitor Windows and SQL Server performance and use Query Waits for specific periods of diagnostics
Integrations	Send alerts to HP Open View via a command line program (opcmsg)
Type	In-House Use
Number of monitored servers	274
Services and Repository installed on same machine	<ul style="list-style-type: none"> • 4 quad core processors • 128 GB of memory • Windows Server 2008 R2
Data storage and grooming options	<ul style="list-style-type: none"> • Repository size: 1.5 TB • Retain 365 days of history
Monitoring features	Monitor Windows and SQL Server performance

Install SQL Diagnostic Manager in silent mode

In addition to the Setup Wizard, SQL Diagnostic Manager allows you to use the Microsoft installer in silent or quiet mode and to simplify SQL Diagnostic Manager deployment.

1. Open an elevated command prompt.
2. Enter the following command-line:
`"PathToSQLDiagnosticManager-x64_c.exe" /v"/L*v "PathToSQLdmInstall.log" SETUPTYPE="Console" /quiet"`


Example

```
"\\SQLdmServerNameHere\c$\Program Files\Idera\SQLdm x64 Installation  
Kit\x64\SQLDiagnosticManager-x64_c.exe" /v"/L*v C:  
\Windows\Temp\IderaSDMInstall.log SETUPTYPE="Console" /quiet"
```

If you want to specify a different installation directory, enter the following command:


```
"PathToSQLDiagnosticManager-x64_c.exe" /v"/L*v "PathToSQLdmInstall.log"  
INSTALLDIR=""PathToInstallDirectory"" SETUPTYPE="Console" /quiet"
```

3. Launch SQL Diagnostic Manager as an administrator.

 After the installation is completed, configure the executable to always run as administrator, review [Run your program as an administrator](#).

How to install the IDERA Dashboard and SQL Diagnostic Manager

This procedure guides you through the installation of the IDERA Dashboard and SQL Diagnostic Manager.

 To perform a silent mode installation, see [Install SQL Diagnostic Manager in silent mode](#).

Start your IDERA SQL Diagnostic Manager installation

You can install the IDERA Dashboard and SQL Diagnostic Manager on any computer that meets or exceeds the product requirements.

To start installing the IDERA Dashboard and SQL Diagnostic Manager:

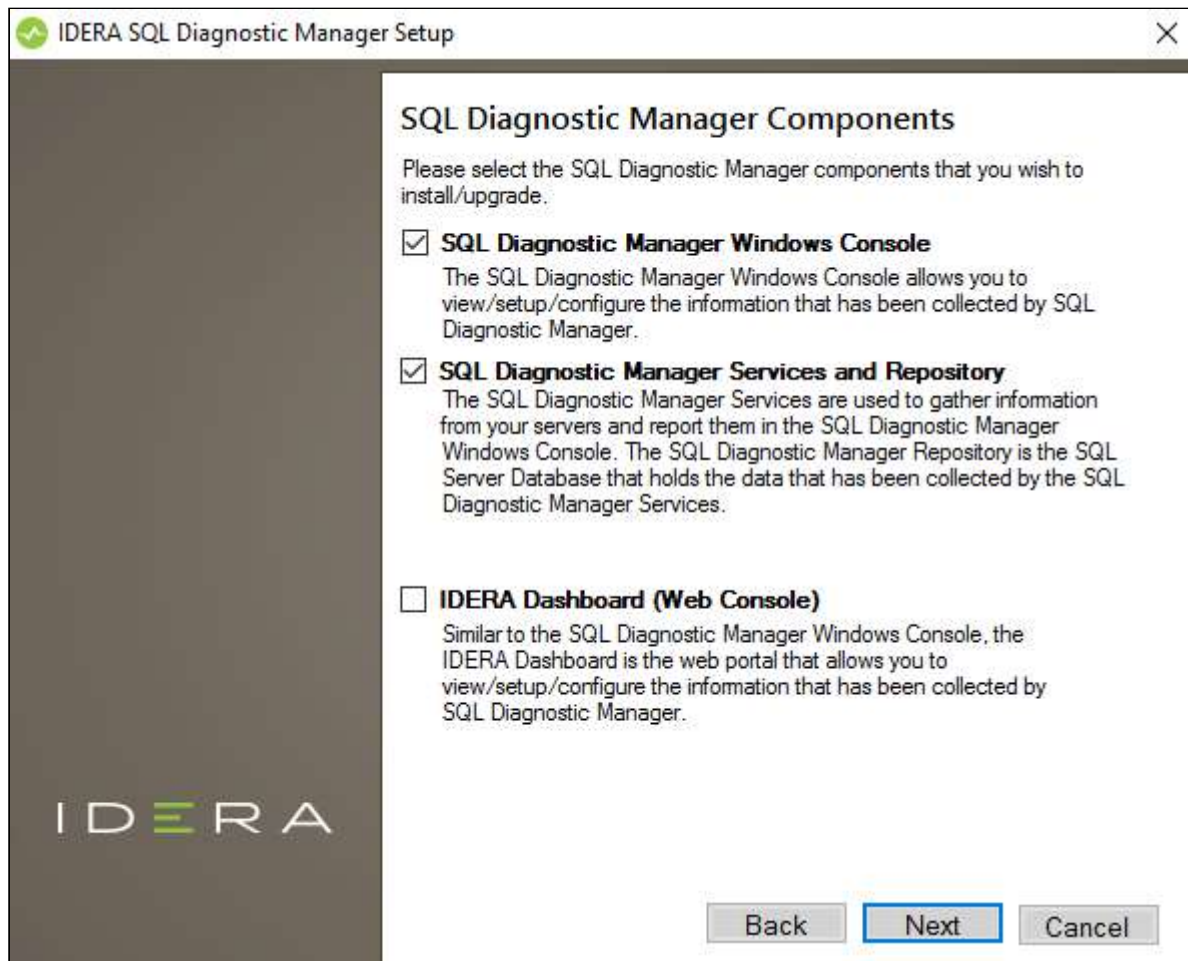
1. Log on with an administrator account to the computer on which you want to install SQL Diagnostic Manager.
2. Run Setup.exe in the root of the installation kit.
3. Click **Install SQL Diagnostic Manager** on the **Quick Start** window.
4. On the **Welcome** window of the setup program, click **Next**.



5. Review the license agreement. To accept this license agreement, click **I accept the terms and conditions of this License Agreement** and then click **Next**.



6. Choose the components you want to install by selecting the respective check boxes. By default, the setup program installs SQL Diagnostic Manager Windows Console and SQL Diagnostic Manager Services and Repository, allowing you to immediately begin monitoring your SQL Server environment. Select the IDERA Dashboard checkbox if you want a complete installation, by default the Web Console is not selected.



i By default, the Management Service, Collection Service, Predictive Service, and Rest Service are installed on the local computer where you are running the setup program.

7. Click **Next**.

Select a Windows user account to use for the service account

SQL Diagnostic Manager uses the service account to communicate between the monitored SQL Server instance and the SQL Diagnostic Manager Repository. The IDERA Dashboard uses this account to connect, discover, and gather configuration information from SQL Servers in your Business environment. The installer grants the "Log on as a Service" right to the account that you specify.

To specify a service account:

1. On the Service Account window, type appropriate credentials in the provided fields under which the services run.



The screenshot shows a window titled "Idera SQL Diagnostic Manager Setup" with a close button in the top right corner. The window is divided into two main sections. On the left is a dark grey vertical bar containing the "IDERA" logo in white. The right section has a white background and is titled "Service Account". Below the title is a paragraph of text: "Specify the service account(s) for IDERA Dashboard and SQL Diagnostic Manager. The service accounts will be used to collect information and to log into the applications. You can log into the IDERA Dashboard to create additional users." Below this text is a checked checkbox with the label "Use the same account for IDERA Dashboard and SQL DM". Underneath the checkbox are two text input fields. The first is labeled "Domain \ UserName:" and contains the text "W12R2\Administrator". The second is labeled "Password:" and contains a series of asterisks. At the bottom right of the window are three buttons: "Back", "Next", and "Cancel". The "Next" button is highlighted with a blue border.

2. Click **Next**. SQL Diagnostic Manager verifies the connection before continuing.

Select service ports to use for the IDERA Dashboard

The IDERA Dashboard uses ports **9292**, **9290**, **9094**, and **9291** for the web application and dashboard services respectively. However, you can specify different service ports.

To specify a service port:

1. On the Service Ports window, specify the ports you want the web application and dashboard services to use.

2. Click **Next**.

 Verify that the Firewall allows TCP (Transmission Control Protocol) traffic through your specified ports.

Select a SQL Server instance to use for the IDERA Dashboard Repository database

The IDERA Dashboard Repository database is where the collected data, alert information, and alert configuration settings are stored.

To specify the SQL Server instance to install the Repository database on:

1. On the Repository Database screen, specify the host SQL Server instance you want to use for the IDERA Dashboard Repository database.

2. Type the name for the IDERA Dashboard Repository database.

3. If your SQL Server environment uses SQL Server Authentication, select the respective check box and click **Change** to enter SQL Server credentials. By default, the IDERA Dashboard uses your Windows credentials when connecting to the Repository database.

i If you use Microsoft SQL Server Authentication, a new window displays where you can type the respective SQL Server login and password.

4. Click **Next**.

Select a SQL Server instance to use for the SQL Diagnostic Manager Repository database

The SQL Diagnostic Manager Repository is where the collected data, alert information, and alert configuration settings are stored.

To specify the SQL Server instance to install the SQL Diagnostic Manager Repository on:

1. Type the host name and instance in the space provided.
2. If your SQL Server environment uses SQL Server Authentication, select the respective check box and click **Change** to enter SQL Server credentials. By default, SQL Diagnostic Manager uses your Windows credentials when connecting to the Repository.
3. If want the SQL Diagnostic Management Service to use SQL Server Authentication , select the respective check box and click **Change** to enter SQL Server credentials. By default, SQL Diagnostic Manager uses your

Windows credentials when connecting to the Repository.

4. Click **Next**.

- ✓ SQL Diagnostic Manager 10.3 requires Microsoft SQL Server 2005 or above running on the computer that hosts the SQL Diagnostic Manager Repository database for all installations.

Select an installation location and type a display name for your instance

You can use the default install location or specify a different path. For your first install, we recommend using the default location. On the Display Name text box, type a unique name for your SQLDM instance. The display name helps you to distinguish different instances of the same product. Click **Next** to continue with the installation.

IDERA SQL Diagnostic Manager Setup

Choose a destination folder

Please provide us the destination folder where you want to install the component(s):

IDERA Dashboard:

SQL DM:

Install this application for:

Anyone who uses this computer (all users)

Only for me (Current User)

Provide a unique name for the SQL DM instance

Type a unique name for your SQL Diagnostic Manager instance to be displayed in the IDERA Dashboard, then click "Check Availability". If the name is available, click Next and proceed with installation.

Display Name:

Note: Display name is available

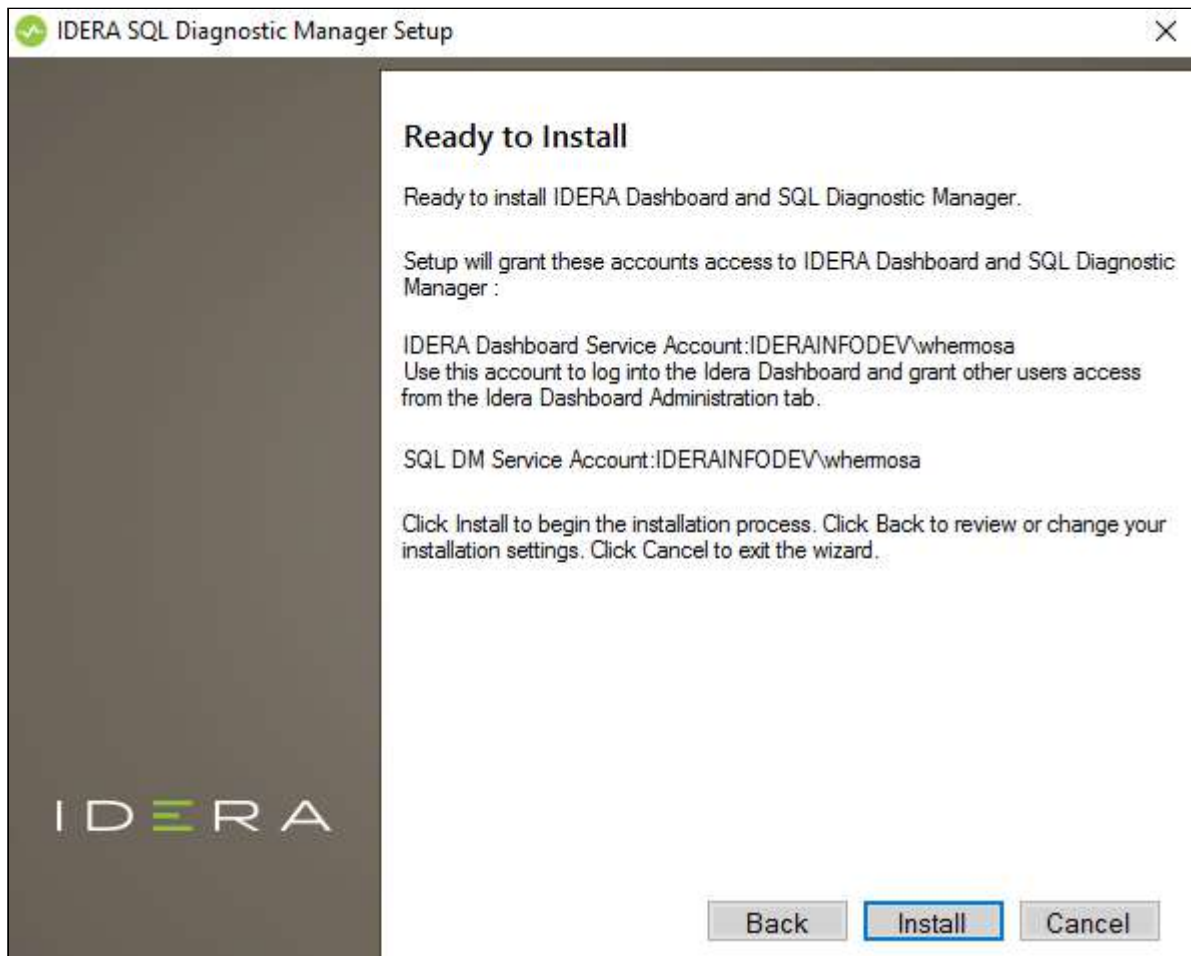
Display name

Provide a unique *display name* for the registered SQL Diagnostic Manager instance on the IDERA Dashboard or the installation process does not complete. Use the **Check Availability** option to verify the Display Name is available.

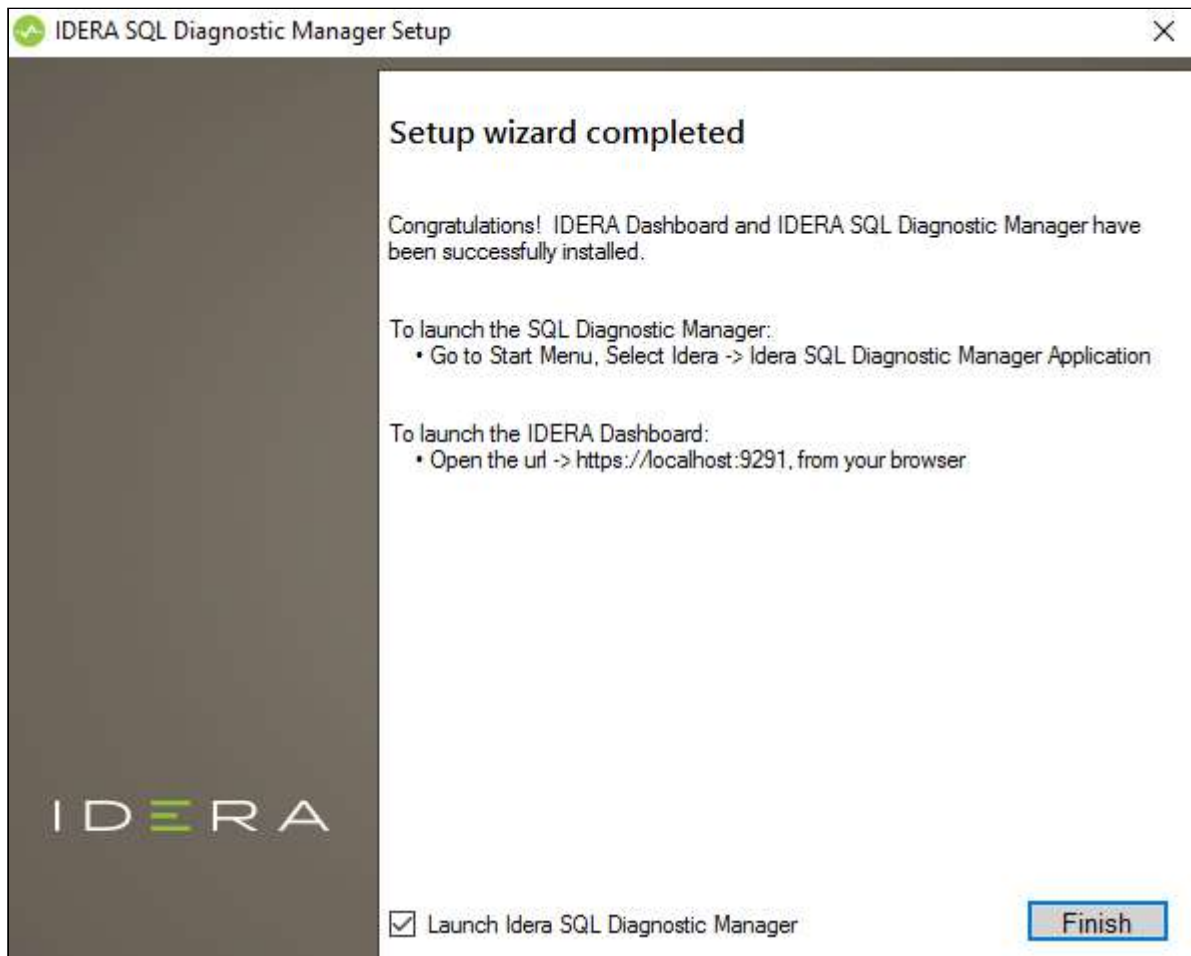
Complete the install

The setup will grant the provided accounts access to SQL Diagnostic Manager and the IDERA Dashboard. The user that installs the IDERA Dashboard has access to this application by default as a Dashboard Administrator. You can choose whether you want other users to have access to the IDERA Dashboard in the Administration view.

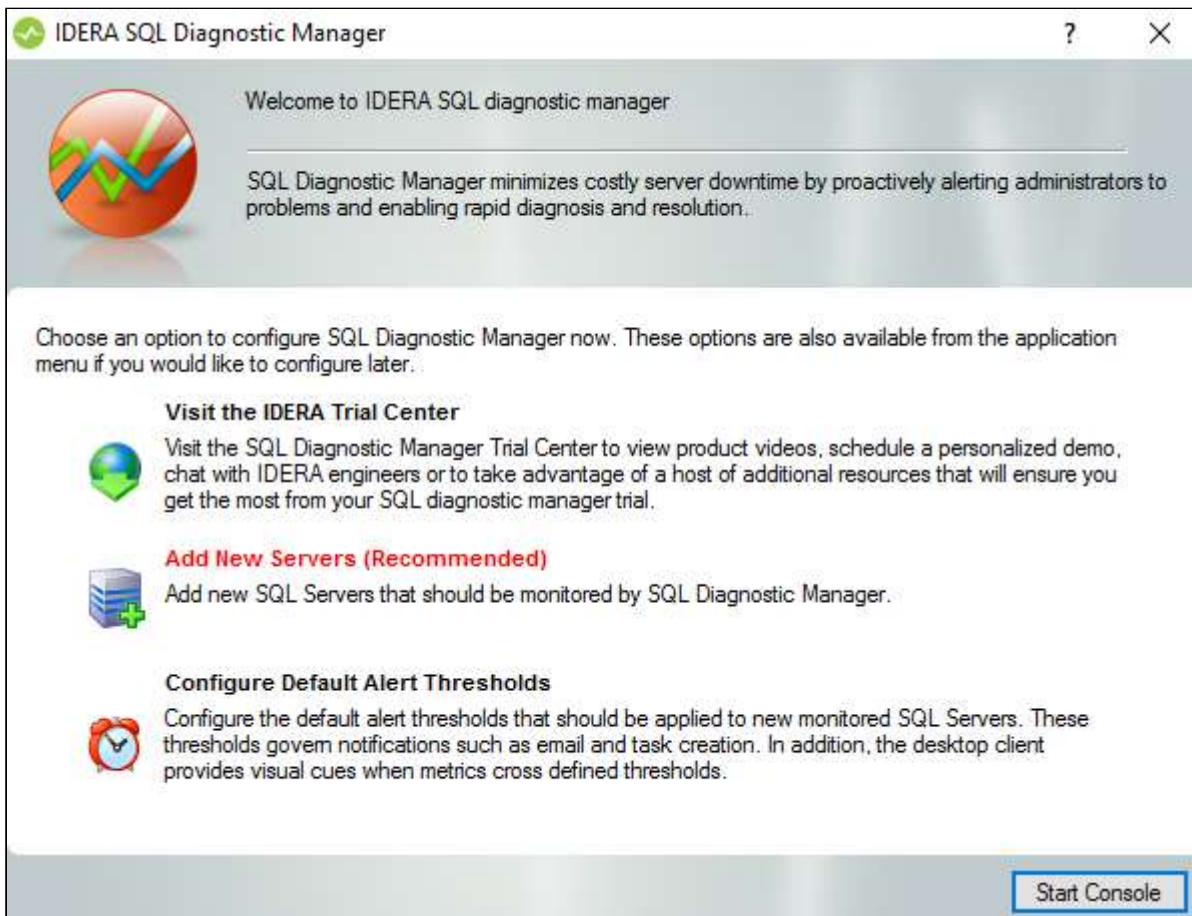
1. Click **Install** to indicate that you are ready to complete your installation with the configurations you specified. If you want to review or change any of your settings, click **Back**.



2. Click **Finish** to exit the IDERA Dashboard Setup Wizard.



When you finish the installation, the Welcome window opens allowing you to **Visit the IDERA Trial Center**, **Add New Servers (Recommended)**, **Configure Default Alert Thresholds**, or **Start the Desktop Console**.



To access the IDERA Dashboard and the SQL Diagnostic Manager web console, see [Log in the IDERA Dashboard and web console](#).

Uninstalling the IDERA Dashboard

If you wish to uninstall the IDERA Dashboard, make sure to unregister all products by clicking the **Manage Products** link of the Products widget in the Administration view. For additional information, see [Managing products in the IDERA Dashboard](#).

Decide whether you want to deploy SQLDM Mobile

SQLDM Mobile includes the IDERA Newsfeed and several dashboard views that define your server health and expose critical alerts. You can access the SQLDM Mobile Web application from most smartphones and mobile devices. The IDERA Newsfeed is a revolutionary new way for DBAs and managers to collaborate, share knowledge, and keep close tabs on your most critical SQL Server issues. With the IDERA Newsfeed, you can quickly share information to stay informed, be more productive and shorten the time to problem resolution. For more information, review the SQLDM Mobile [architecture and components](#), as well as its [requirements](#).

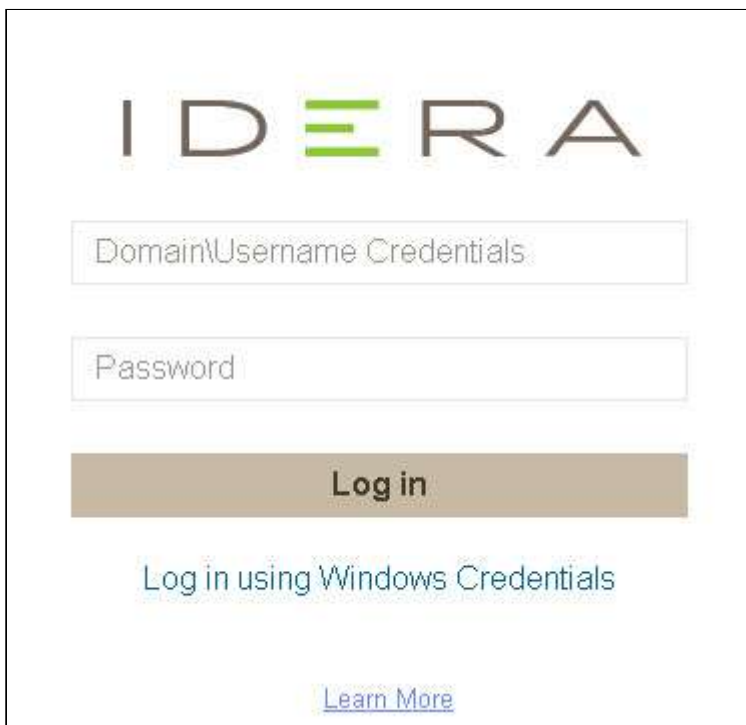
Log in the IDERA Dashboard and web console

Once you have installed and configured your IDERA Dashboard and SQL Diagnostic Manager deployments, you can login to the web console by doing the following:

1. Open your selected Browser, make sure it is compatible with the IDERA Dashboard console requirements.
2. Type the IDERA Dashboard product URL: **http://<machinename>:<port>** where **<machinename>** is the name of your host or machine, and **<port>** is the port specified during installation. The default URL is **https://<localhost>:9291** or **https://<machinename>:9291**.

✔ The port 9290 is used for http protocol, while the port 9291 is used for https protocol.

3. When the IDERA Dashboard web console launches on your browser, use your Windows user account **<domain\user>** with the respective password to log into the product.



⚠ The IDERA Dashboard Web Application service comes with SSL already set up. For more information on running the IDERA Dashboard over SSL, see [Run the IDERA Dashboard over SSL \(HTTPS\)](#)

Deploy SQLDM Mobile or IDERA Newsfeed

SQLDM Mobile is a Web-based application that includes the IDERA Newsfeed technology and several dashboard views that define your server health and expose critical alerts. You can access the SQLdm Mobile Web site from most smartphones and mobile devices.

Use the following links to prepare for your SQLDM Mobile or IDERA Newsfeed deployment:

- Check the [supported installation scenarios](#)
- Learn about the [components and architecture](#)
- Review [system requirements](#)
- View the [installation instructions](#)

SQLDM Mobile components and architecture including IDERA Newsfeed

SQLDM Mobile is an easy-to-use, Web-based application that seamlessly integrates into your existing diagnostic and monitoring workflow. The SQLDM Mobile Web site includes many metrics and alerts panels that let you quickly ascertain performance issues in your environment.

SQLDM Mobile also includes the IDERA Newsfeed technology. The Newsfeed provides an easy-to-use, social-media-based collaboration tool you can use to solve issues faster while keeping your team apprised of changing server statuses.

Whether you are tied to your desktop, on the go, or managing remote servers, these features can fit your unique monitoring needs.

The SQLDM Mobile architecture easily runs in your SQL Server environment with minimal configuration. All components run outside and separate from SQL Server processes, and do not add to or modify any of your native SQL Server files or services.

Components

IDERA Newsfeed for SQL Diagnostic Manager

The IDERA Newsfeed is a user interface that seamlessly integrates with the SQL Diagnostic Manager Console, letting you monitor your instances and collaborate with coworkers at the same time, in real time.

IDERA Newsfeed Platform

The IDERA Newsfeed Platform consists of the SQLDM Mobile Repository database and the SQLDM Mobile and Newsfeed Service.

SQLDM Mobile Repository database

The SQLDM Mobile Repository (Mobile Repository) is a central database that stores your configuration settings, comments and other activity on stories, as well as notifications. Consider deploying the Mobile Repository to the same computer that hosts your SQL Diagnostic Manager Repository.

SQLDM Mobile and Newsfeed Service

The SQLDM Mobile and Newsfeed Service facilitates communications between the SQLDM Mobile Repository and your SQL Diagnostic Manager application. By default, this service is installed on the same computer where you run the SQLDM Mobile and Newsfeed setup program. Consider SQLDM Mobile and Newsfeed Service to the same computer that hosts your SQL Diagnostic Manager services, such as the SQLDM Management Service.

Action Provider

The IDERA Newsfeed Action Provider converts SQL Diagnostic Manager alerts about a server status and performance metrics into notification emails that SQLDM Mobile & Newsfeed Service can send to the appropriate people. The setup program installs the action provider on the computer hosting your SQL Diagnostic Manager Repository.

Authentication Provider

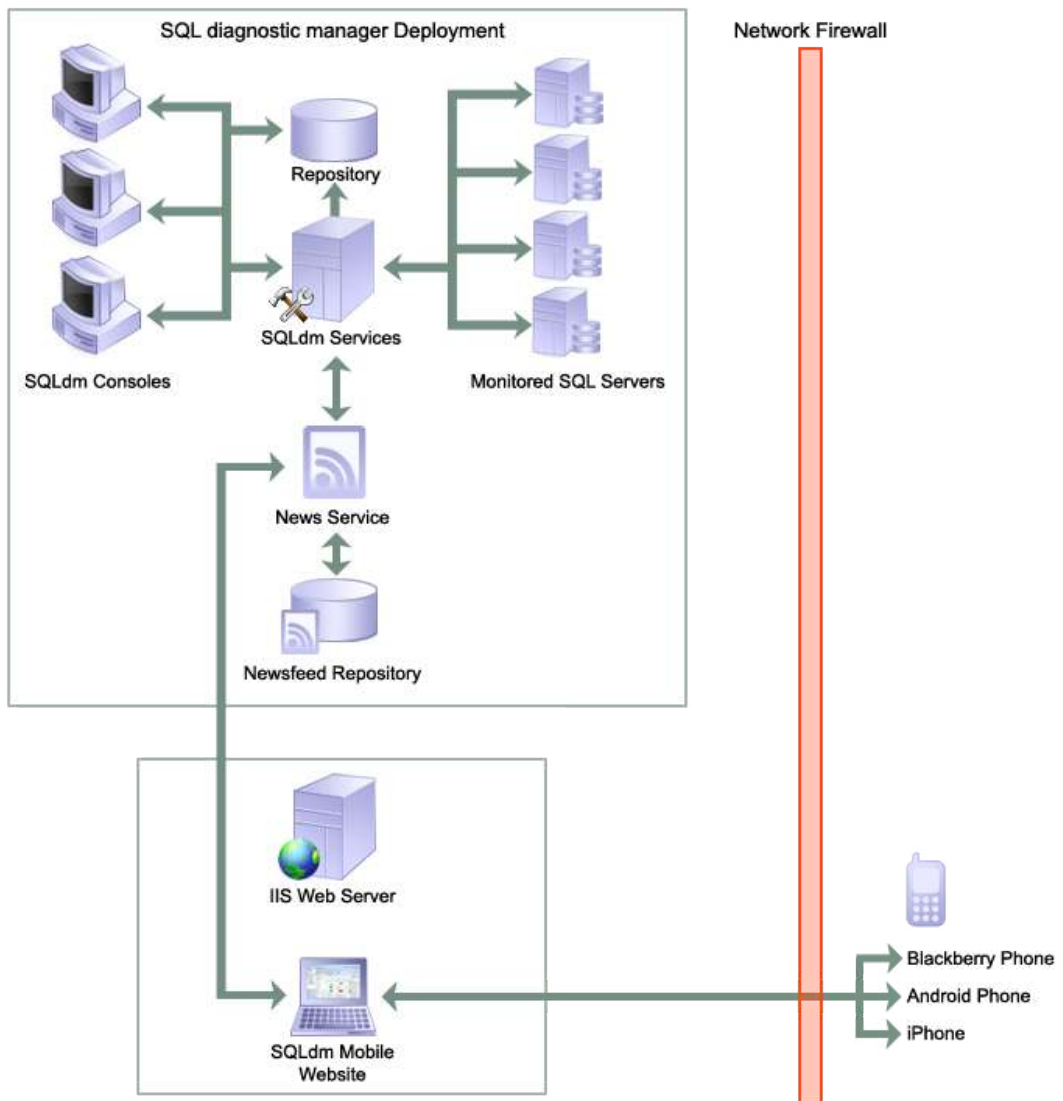
The IDERA Newsfeed Authentication Provider is the SQL Diagnostic Manager application that the IDERA Newsfeed Platform uses to authenticate a user when he or she signs up or logs on. The IDERA Newsfeed

Platform leverages the application security settings as defined in SQL Diagnostic Manager, determining which monitored SQL Server instances a given user can access.

SQLDM Mobile Web application

The SQLDM Mobile website provides access to the same monitoring and collaboration features of the IDERA Newsfeed technology via a mobile device, such as a smartphone or iPad. The SQLDM Mobile website also includes SQL Diagnostic Manager panels so you can quickly check server performance when you are away from your desk.

Architecture



SQLDM Mobile and IDERA Newsfeed requirements

You can easily install SQLDM Mobile on any computer that meets or exceeds the following requirements. The SQLDM Mobile installation includes the IDERA Newsfeed technology and these related components:

- SQLDM Mobile Web application
- SQLDM Mobile & Newsfeed Service
- SQLDM Mobile Repository

For more information, review all the SQLDM Mobile [components and architecture](#).

General requirements for all components

All SQLDM Mobile components require Microsoft .NET Framework version 4.0 or later.

Permission requirements

The SQLDM Mobile and Newsfeed Service account requires the following permissions. By default, setup program assigns the Local System account to the SQLDM Mobile and Newsfeed Service. During install, you can input credentials for a Windows user account or SQL Server login.

Permission	Why it's required
Log on as Service	Allows the SQLDM Mobile and Newsfeed Service account to run as a service.
Read and write privileges on the SQLDM Mobile Repository database	Allows the SQLDM Mobile and Newsfeed Service to receive and maintain performance and configuration data in the SQLDM Mobile Repository.

Port requirements

By default, the SQLDM Mobile components use the following ports.

Port Number	Where	Which component uses it	Why
25	Mail Server	SQLDM Mobile and Newsfeed Service	Allows the SQLDM Mobile and Newsfeed Service to send notification emails.
80	Web Server hosting the SQLDM Mobile Website	SQLDM Mobile Web application	Allows access to the SQLDM Mobile screens via a mobile device inside or outside your corporate firewall. By default, IIS uses port 80 for the local virtual directory, so check your Web Server settings to pick an appropriate port for SQLDM Mobile. You can specify a different port during install or later when you need to change this configuration.

Port Number	Where	Which component uses it	Why
5166, 5167	Computer running the SQL Diagnostic Manger services	SQL Diagnostic Manager services	Allow communications between SQL Diagnostic Manager and the IDERA Newsfeed Platform for signup and login authentication.
5168	Computer running the SQLDM Mobile and Newsfeed Service	SQLDM Mobile and Newsfeed Service	Allows communications between the IDERA Newsfeed and SQL Diagnostic Manager for server status updates.

Mobile device requirements

You can access SQLDM Mobile on any smart phone or other mobile device that meets these minimum operating system requirements:

- Android version 2.1 or later
- iOS version 3.2 or later
- RIM (BlackBerry OS) version 6.0 or later
- Windows Phone 7

You can run the SQLDM Mobile Web application in a native browser that supports HTML 5, such as Google Chrome, Mozilla Firefox, or Edge Browser. SQLDM Mobile does not support Microsoft Internet Explorer 10+ or earlier.

- ✔ If you plan to use IE 9 or later to access the SQLDM Mobile Web site, make sure that you disable the **Display intranet sites in Compatibility View** setting. To change this setting, click **Compatibility View Settings** on the **Tools** menu in the IE browser window.

Web server requirements

The SQLDM Mobile Web application requires the following Web Server configuration and Services:

Type of requirement	Minimum value
CPU	2.0 GHz
Memory	1 GB
Disk space	21 MB
Microsoft Internet Information Services (IIS)	Version 8.0 IIS 7.2 or later

Type of requirement	Minimum value
Operating System	<ul style="list-style-type: none"> • Windows Server 2008 • Windows 7 • Windows 2008 R2 • Windows 8 • Windows 10 • Windows 2012 • Windows 2016

✔ To successfully run the SQLDM Mobile Web application, make sure that Microsoft .NET Framework version 4.0 or later is installed. Also, verify the following service roles in IIS are enabled:

- Management Tools
 - IIS Management Console
- Web Server
 - Application Development
 - ASP.NET (4.0 or later)
 - ISAPI Extensions
 - ISAPI Filters
 - .NET Extensibility (4.0 or later)
 - Common HTTP Features
 - Default Document
 - Directory Browsing
 - HTTP Errors
 - Static Content
 - Health and Diagnostics
 - HTTP Logging
 - Performance
 - Static Content Compression
 - Security
 - Request Filtering

Repository requirements

You can install the SQLDM Mobile Repository on a computer running the following versions of SQL Server software:

- SQL Server 2005 SP1+
- SQL Server 2008
- SQL Server 2008 R2
- SQL Server 2012
- SQL Server 2014
- SQL Server 2016
- SQL Server 2017
- SQL Server 2019

Keep in mind that the Repository database and corresponding SQLDM Mobile & Newsfeed Service require 19 MB of disk space.

How to install SQLDM Mobile and IDERA Newsfeed

This procedure guides you through the installation of SQLDM Mobile and Newsfeed. Before installation, review the associated [architecture and components](#). Although you can install different components on different computers, this procedure recommends the following deployments:

- Install the Mobile and Newsfeed components on the same computer where your SQL Diagnostic Manager services reside
- Install the Mobile web components on your corporate Web Server

✔ In larger environments that contain more than one SQL Diagnostic Manager installation, consider dedicating a SQLDM Mobile deployment per each SQL Diagnostic Manager application.

✔ If you are interested in installing only the IDERA Newsfeed without the additional components used with SQLDM Mobile, see [How to install IDERA Newsfeed only](#).

Install or upgrade to SQL Diagnostic Manager 10.4

SQLDM Mobile and the IDERA Newsfeed technology are supported by SQL Diagnostic Manager 7.0 or later versions. To successfully install and deploy SQLDM Mobile, [Install](#) or [upgrade](#) first to SQL Diagnostic Manager 10.3

Start your SQLDM Mobile and Newsfeed install

You can install the SQLDM Mobile components on any computer that meets or exceeds the [product requirements](#). By default, the SQLDM Mobile and Newsfeed setup program installs the SQLDM Mobile and Newsfeed Service to the local computer.

To start installing SQLDM Mobile and Newsfeed:

1. Log on with an administrator account to the computer on which you want to install the SQLDM Mobile and Newsfeed Service. We recommend that you install this service on the same computer that already hosts your SQL Diagnostic Manager services.
2. Run Setup.exe in the root of the SQL Diagnostic Manager installation kit.
3. Click **SQLDM Mobile and Newsfeed** on the **Quick Start** window.
4. On the **Welcome** window of the setup program, click **Next**.
5. Review the license agreement. To accept this license agreement, click **I accept the terms in the license agreement** and then click **Next**.

Begin configuring your install

We recommend installing the Mobile and Newsfeed components in your SQL Diagnostic Manager environment, and then re-running the setup program to install the SQLDM Mobile Web application on your Web Server. You can use the default install location or specify a different path. For your first install, we recommend using the default location. By default, the SQLDM Mobile and Newsfeed Service is also installed in the same location.

On the **Custom Setup** window, choose the **Mobile and Newsfeed components**, and then click **Next**.

Add the Windows credentials to use for the SQLDM Mobile and Newsfeed Service account

You can specify the Local System account or a Windows user account. Make sure that the specified account includes the [required permissions](#).

To specify a service account:

1. Select the Local System account or type the appropriate credentials in the provided fields.
2. Click **Test Credentials** to validate the account.
3. Click **Next**.

Select the SQL Server instance to use for the Repository database

Store the published status updates and other configuration settings in the SQLDM Mobile Repository.

- ✔ SQLDM Mobile requires Microsoft SQL Server 2005 or above running on the computer that hosts the Mobile Repository database.

To specify on which SQL Server instance you want to install the SQL Diagnostic Manager Repository:

1. Select the SQL Server instance you want to use. The SQL Server menu lists SQL Server instances provided by your domain controller. ***If you do not see your SQL Server instance in the list***, type the host name and instance in the space provided.
2. Click **Next**.

Add the authentication credentials to create the Repository database

You can use either Windows or SQL Authentication to create the Mobile Repository database. Make sure the specified Windows account includes the [required permissions](#).

To specify a service account:

1. Choose the authentication that you want to use, and then type the appropriate credentials in the provided fields.
2. Click **Test Connection** to validate the account can connect to the specified SQL Server instance.
3. Click **Next**.

Add the authentication credentials to connect to the Repository database at run time

You can use either Windows or SQL Authentication to connect to the Mobile Repository database. Make sure the specified Windows account includes the [required permissions](#).

To specify a service account:

1. Choose the authentication that you want to use, and then type the appropriate credentials in the provided fields.
2. Click **Test Connection** to validate the account can connect to the specified SQL Server instance.
3. Click **Next**.

Select a location for the SQLDM Management Service

SQLDM Mobile uses its own Authentication Provider to enforce application security across your SQL Diagnostic Manager and SQLDM Mobile deployments.

On the **Authentication Provider** window, specify the computer that hosts the SQLDM Management Service, and then click **Next**.

Complete the Mobile & Newsfeed components install

Click **Install** to indicate that you are ready to complete your install with the configurations you specified.

Begin the SQLDM Mobile Web components install

We recommend installing the SQLDM Mobile Web components on your corporate Web Server. Make sure your Web Server meets the [software and port requirements](#).

To start installing the SQLDM Mobile Web application:

1. Log on with an administrator account to the computer on which you want to install the SQL Diagnostic Manager Web application. We recommend that you install this application on the same Web Server that already hosts your corporate Web site.
2. Restart the SQLDM Mobile and Newsfeed setup program and navigate to the **Custom Setup** window.
3. Choose the **Mobile Web components**, and then click **Next**.

Select the Web Server settings you want the SQLDM Mobile Web application to use

The setup program installs the SQLDM Mobile Web application in a unique virtual directory on your Web Server computer. You can install the SQLDM Mobile Web application as a new Web site or add it to an existing Web site, such as your corporate intranet.

Specify the name of the computer that hosts the SQLDM Mobile and Newsfeed Service so that you can send communications between the SQLDM Mobile components and SQLDM.

Make sure your Web Server meets the SQLDM Mobile [requirements](#).

To configure the Web Server settings:

1. Choose whether to install the SQLDM Mobile Web application as a new Web site.
2. Specify the name of the computer on which you installed the SQLDM Mobile and Newsfeed Service.
3. Click **Next**.

Complete the Mobile Web components install

Click **Install** to indicate that you are ready to complete your install with the configurations you specified.

How to install IDERA Newsfeed only

This procedure guides you through the installation of IDERA Newsfeed only and none of the additional components used with SQLDM Mobile. Use this solution in a larger environment where your SQL Server team or environment is spread across multiple locations and you need an easier, more streamlined way to troubleshoot issues. The IDERA Newsfeed can help you capture or bookmark the team resolutions for future reference. You can implement this option without need to allot the additional hardware and IIS resources necessary to deploy a complete, mobile-available solution.

Before installing Newsfeed, review the associated [architecture and components](#).

Install or upgrade to SQL Diagnostic Manager 10.6

The IDERA Newsfeed technology is supported by SQL Diagnostic Manager 7.0 and above versions. To successfully install and deploy IDERA Newsfeed, first [install](#) or [upgrade](#) to SQL Diagnostic Manager 10.4.

Start your IDERA Newsfeed install

Before installing Newsfeed, review the associated [architecture and components](#). You can install the IDERA Newsfeed on any computer that meets or exceeds the [product requirements](#). By default, the SQLDM Mobile and Newsfeed setup program installs the SQLDM Mobile and Newsfeed Service to the local computer.

To start installing IDERA Newsfeed:

1. Log on with an administrator account to the computer on which you want to install the SQLDM Mobile and Newsfeed Service.
2. Run Setup.exe in the root of the SQL Diagnostic Manager installation kit.
3. Click **SQLDM Mobile and Newsfeed** on the **Quick Start** window.
4. On the **Welcome** window of the SQLDM Mobile and Newsfeed setup program, click **Next**.
5. Review the license agreement. To accept this license agreement, click **I accept the terms in the license agreement**, and then click **Next**.

Begin configuring your install

You can use the default install location or specify a different path. For your first install, we recommend using the default location. By default, the SQLDM Mobile and Newsfeed Service is also installed in the same location.

On the **Custom Setup** window, choose the **Mobile and Newsfeed components**, and then click **Next**.

Add the Windows credentials to use for the SQLDM Mobile and Newsfeed Service account

You can specify the Local System account or a Windows user account. Make sure the specified account includes the [required permissions](#).

To specify a service account:

1. Select the Local System account or type the appropriate credentials in the provided fields.
2. Click **Test Credentials** to validate the account.
3. Click **Next**.

Select the SQL Server instance to use for the Repository database

The IDERA Newsfeed Repository is where the published status updates and other configuration settings are stored.

- ✔ SQLdm Mobile requires Microsoft SQL Server 2005 or above running on the computer that hosts the Newsfeed Repository database.

To specify the SQL Server instance to install the SQL Diagnostic Manager Repository on:

1. Select the SQL Server instance you want to use. The SQL Server menu lists SQL Server instances provided by your domain controller. **If you do not see your SQL Server instance in the list**, type the host name and instance in the space provided.
2. Click **Next**.

Add the authentication credentials to create the Repository database

You can use either Windows or SQL Authentication to create the Newsfeed Repository database. Make sure the specified Windows account includes the [required permissions](#).

To specify a service account:

1. Choose the authentication that you want to use, and then type the appropriate credentials in the provided fields.
2. Click **Test Connection** to validate the account can connect to the specified SQL Server instance.
3. Click **Next**.

Add the authentication credentials to connect to the Repository database at run time

You can use either Windows or SQL Authentication to connect to the Newsfeed Repository database. Make sure the specified Windows account includes the [required permissions](#).

To specify a service account:

1. Choose the authentication that you want to use, and then type the appropriate credentials in the provided fields.
2. Click **Test Connection** to validate the account can connect to the specified SQL Server instance.
3. Click **Next**.

Select a location for the SQLDM Management Service

SQLDM Mobile uses its own Authentication Provider to enforce application security across your SQL Diagnostic Manager and IDERA Newsfeed deployments.

On the **Authentication Provider** window, specify the computer that hosts the SQLDM Management Service, and then click **Next**.

Complete the Newsfeed install

Click **Install** to indicate that you are ready to complete your install with the configurations you specified.

How to install SQL Diagnostic Manager Console only

This procedure guides you through the installation of the SQL Diagnostic Manager Console only.

- ✔ You must have a full installation of SQL Diagnostic Manager already running in your environment before attempting to install the SQL Diagnostic Manager desktop client console.

Perform your install

You can install SQL Diagnostic Manager desktop client console on any computer that meets or exceeds the [product requirements](#).

To install SQL Diagnostic Manager Console only:

1. Log on with an administrator account to the computer on which you want to install the SQL Diagnostic Manager desktop client console.
2. Navigate to the location where the SQL Diagnostic Manager Console file resides in the distribution media.
3. **If you use an x64 operating system**, run `SQLDiagnosticManager-x64_c.exe`. **If you use an x86 OS**, run `SQLDiagnosticManager_c.exe`.
4. On the **Welcome** window of the setup program, click **Next**.
5. Review the license agreement. To accept this license agreement, click **I accept the terms in the license agreement**, and then click **Next**.
6. Select the default installation location or specify a different path.
7. Choose whether you want any user or only the current user to access this application, and then click **Next**.
8. Verify that SQL Diagnostic Manager displays **Console Only** as selected, and then click **Next**.
9. Click **Install** to indicate that you are ready to complete your install with the configurations you specified.
10. Click **Finish** to exit the setup program.

Complete the post-installation steps

When you first open the SQL Diagnostic Manager desktop client console, you must complete the fields necessary to [connect to your existing SQL Diagnostic Manager Repository](#). It is important that you have the information available to connect to the proper SQL Server instance and SQL Diagnostic Manager Repository database using the proper authentication credentials.

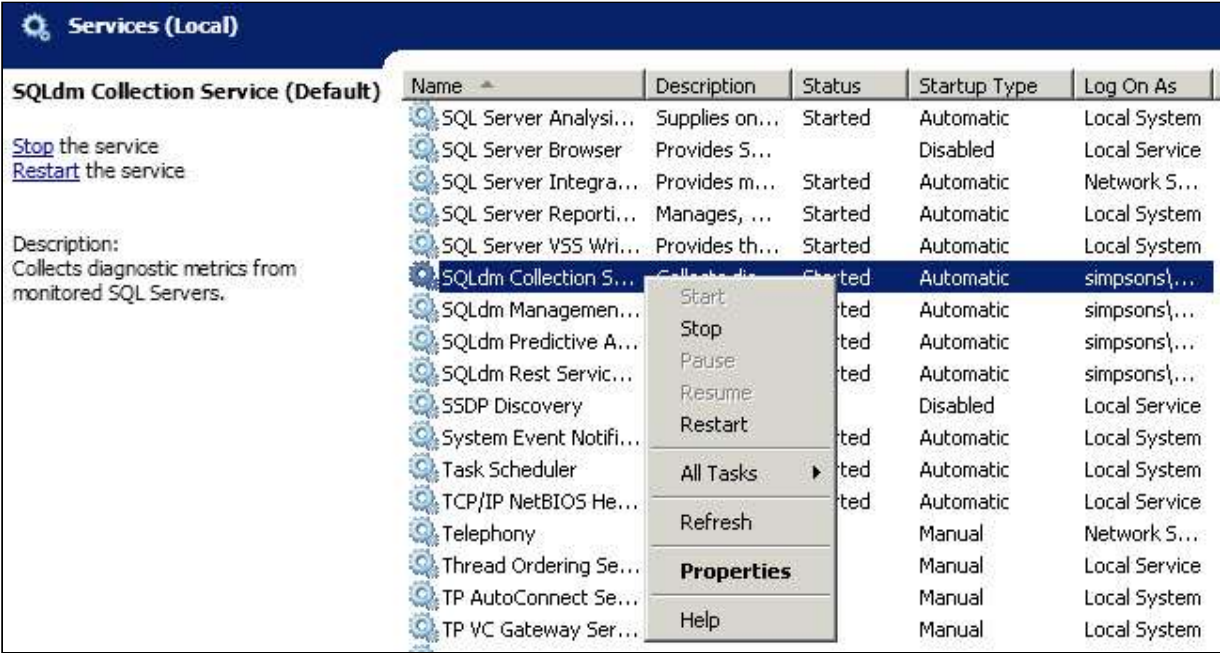
Deploy SQL Diagnostic Manager in a high availability group

This procedure guides you through the installation of SQL Diagnostic Manager in a high availability group. SQL Diagnostic Manager does not support direct installations under this scenario; however, you can deploy the SQL Diagnostic Manager Repository in one of the nodes that belongs to a specific availability group and configure an [availability group listener](#) that serves as a global link between nodes.

- ✓ Make sure to restart the SQL Diagnostic Manager services, apply the newly-generated license using the Manage Licenses button before attempting to apply the license directly using the Management Console or you may encounter issues.

1. Deploy the SQL Diagnostic Manager Repository targeting a node that is part of an availability group. For example, Node 1\Instance A.
2. Move the SQL Diagnostic Manager Repository to an availability group in the primary replica with Listener Y and port 300 (as example). To review how to create an availability group, see Microsoft document [Creating and Configuring a New Availability Group](#).
3. A new license key for Listener Y, Port 300 is required. See [Capturing an availability group's listener name and port designation](#) to send your request to [IDERA Sales](#).
4. Using the SQLdm Management Service Configuration wizard modify the target repository to Listener Y, port 300.

5. Save changes, and then restart the SQL Diagnostic Manager services. This process takes approximately 5 minutes.

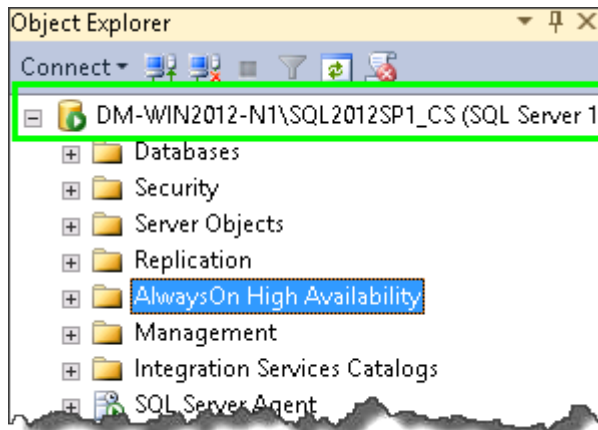


- 6. Upon opening the SQL Diagnostic Manager Console, the system recognizes that you are trying to access the selected node without the proper license and throws an error message.
- 7. Click the **Manage Licenses** button of the error message and type the new license for Listener Y, Port 300. Open the SQL Diagnostic Manager Console, go to **Help > Manage Licenses** in the Toolbar menu, and the License Keys window displays. Type the new license key in the available space, and then click Enter.

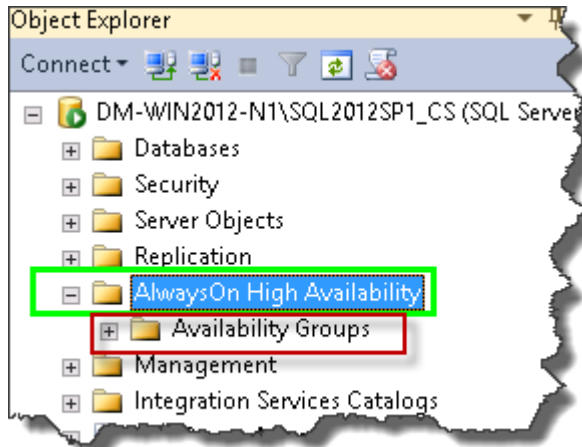
AlwaysOn availability group listener name and port designation

This procedure guides you through the necessary steps to capture an availability group's listener name and port designation. This information is essential for IDERA Sales to generate a special license that allows you to host the SQL Diagnostic Manager Repository in an availability group. As a pre-requisite the SQL Diagnostic Manager Repository needs to be hosted in an availability group within the primary replica.

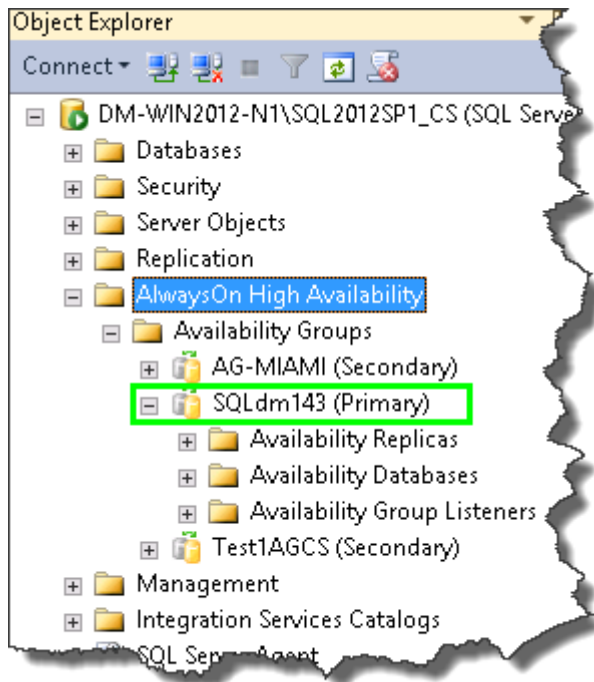
1. Within SSMS, go to node that contains the availability group that hosts the SQL Diagnostic Manager Repository and is the primary replica.



2. Expand the **AlwaysOn High Availability** folder and then the **Availability Groups** Folder.



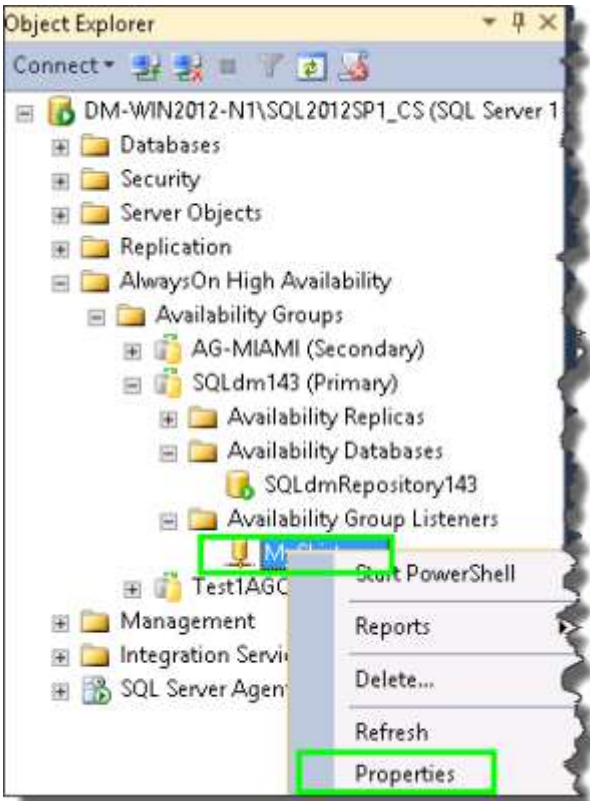
- Expand the availability group that is the primary replica.



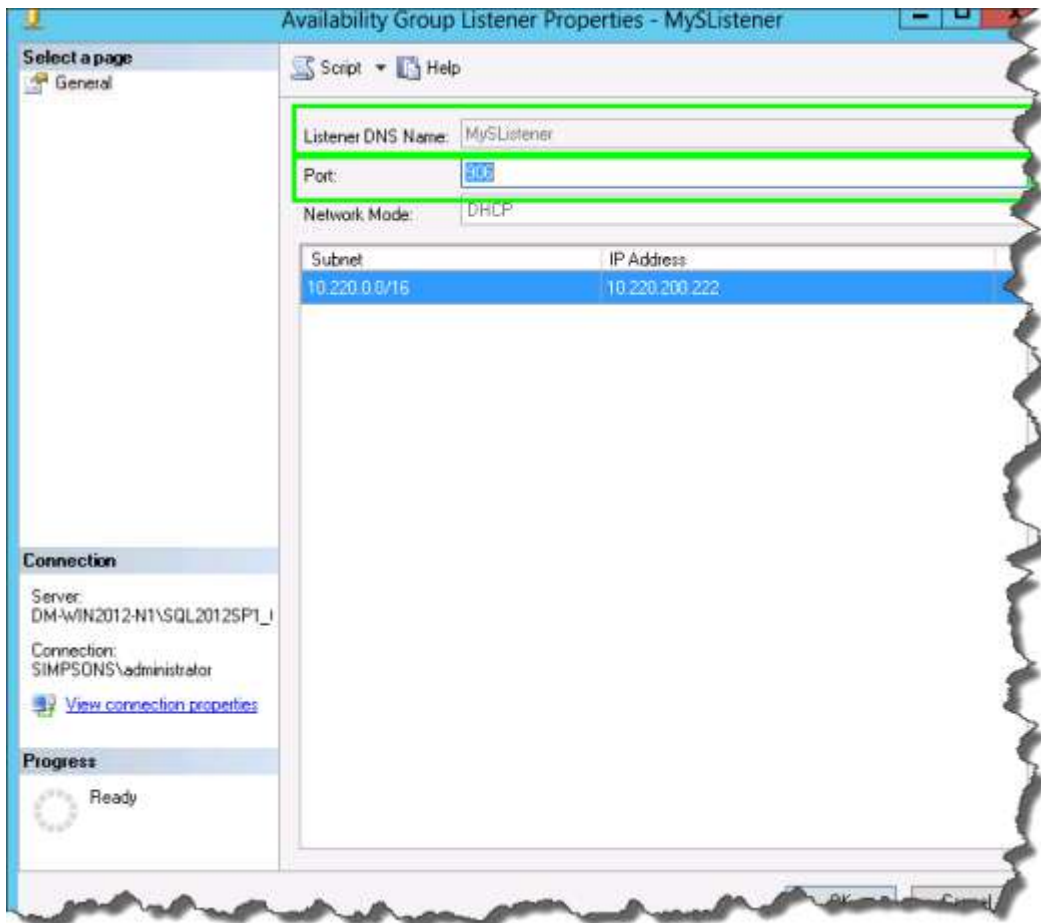
- Expand the **Availability Group Listeners** Folder.



- Right-click the listener, select **Properties**, and the Availability Group Listener Properties dialog Displays.



Send [IDERA Sales](#) the **Listener DNS Name** and **Port designation** information to obtain a new license to host the SQL Diagnostic Manager Repository in an availability group.



How SQL Diagnostic Manager works with a virtual environment

Virtual environments, such as those provided by Hyper-V or VMware, allow you to run multiple instances of computers or "virtual machines" on a single, physical host server. These VMs create a virtual environment that enables IT shops to more fully utilize the hardware they already have, allowing administrators to maintain fewer physical servers which then consume less power, take up less real estate, and generate less heat.

While there are many advantages to using a virtual environment, there are some disadvantages. One is that a virtual machine currently is not as fast or efficient as a standalone physical machine. Administrators must closely monitor and manage their virtual environments because over-utilized hardware can quickly and severely impact the performance of their servers, which are constantly waiting for physical resources to become available to process their workloads.

SQL Diagnostic Manager includes VM monitoring to allow administrators to get a glimpse of what is going on inside the virtual environment in which their SQL Servers are running, to make sure that their VMs and their host machines are running at peak efficiency. This topic is a general outline of how SQL Diagnostic Manager works in a virtual environment and includes links to other areas where you can find additional VM-specific information.

Using the hypervisor to manage resources

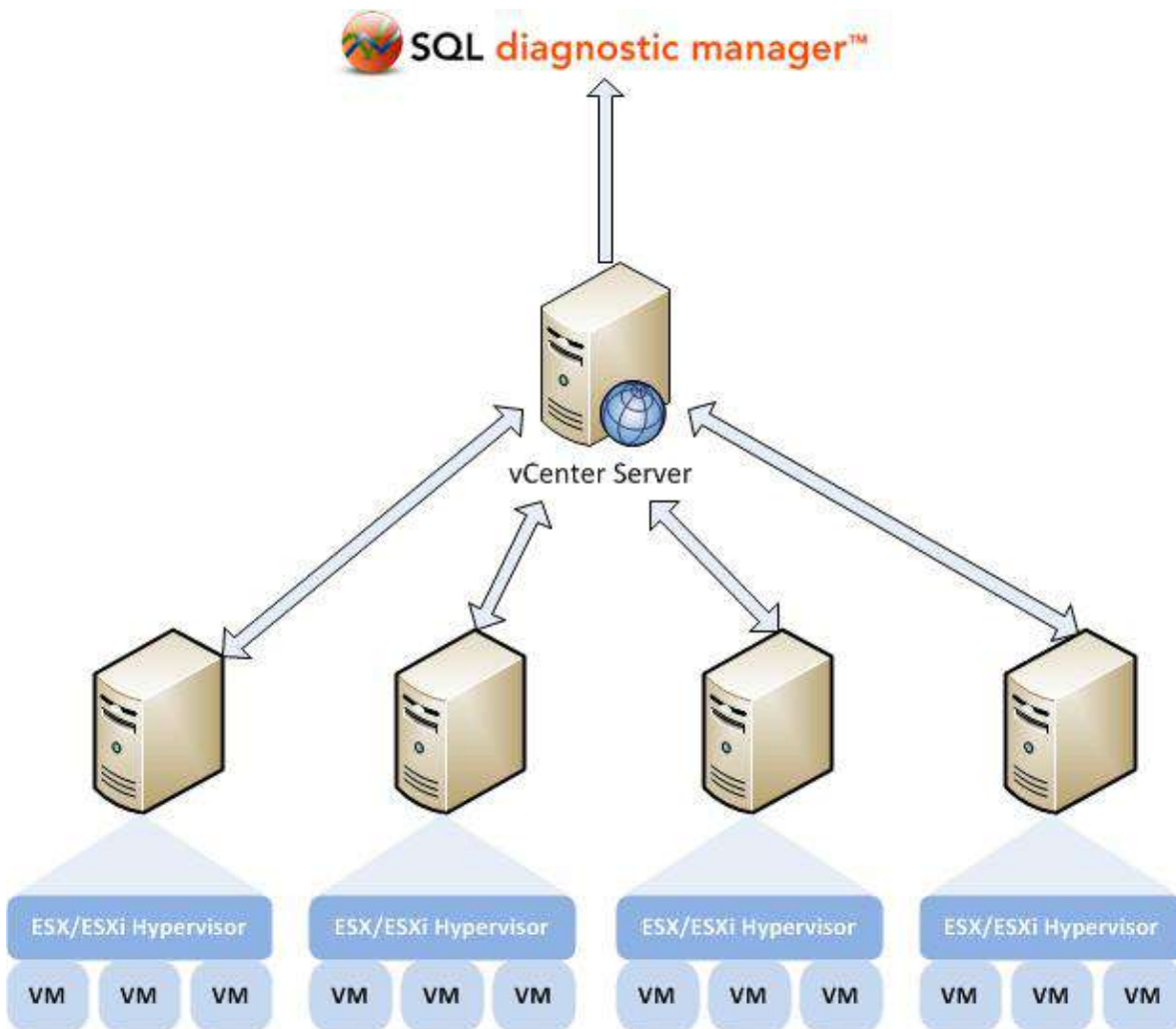
On the physical server hosting a virtual environment, there is a process called the hypervisor or virtualization manager running in the background. The hypervisor manages the interactions between the virtual machines and the physical hardware, manages resources, and then distributes the physical resources as required by each of the VMs it is hosting.

An administrator can intentionally over commit resources on a physical host with the idea that it is unlikely that all of the VMs need to utilize all of the resources on the host at the same time. The hypervisor is also affected by this shift, and manages any occasion when all of the active VMs need more physical resources than is available from the host. Note that there are ramifications for the extra processing that is occurring when the hypervisor is taxed. This additional processing is the overhead for a virtual environment and consumes CPU, memory, and network resources.

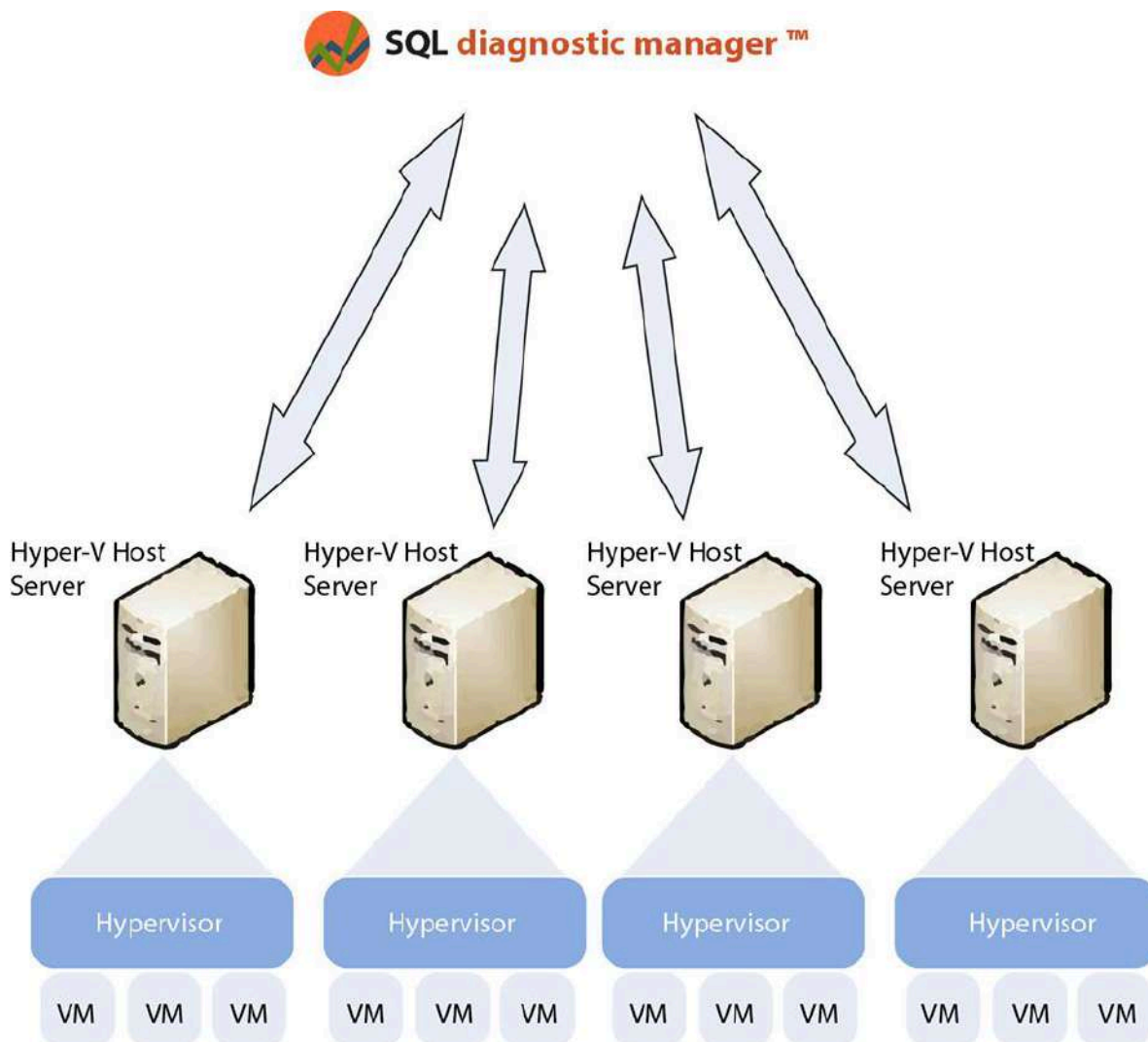
How SQL Diagnostic Manager collects and presents metrics involving a virtual environment

The guest operating system on a VM operates no differently than it does when running on a physical machine with full access to all of the resources at its disposal. But the machine hosting the VM operates differently because of the virtual environment. The physical resources of the host are shared with the other VMs on the host. Due to the additional overhead of virtualization, and the hypervisor managing contention for resources, the metrics that the guest OS reads as utilization of its resources are not entirely accurate as they do not take into account that virtualization overhead, nor does it measure that it is waiting on physical resources to process its workload.

The image below shows a typical virtualized environment with VMware. In this environment, the ESX/ESXi hypervisor on each individual host server collects the performance statistics and reports the data back to a central vCenter Server. SQL Diagnostic Manager then collects that information from the vCenter Server in order to show those statistics along with the other SQL related performance information.



In virtualized environment with Hyper-V, the process of how SQL Diagnostic Manager collects and presents metrics is slightly different from that of a VMware setup. In this environment, SQL Diagnostic Manager collects all the required performance statistics from the Hyper-V host server and virtual machines using the WMI interface. See [Hyper-V user requirements](#) for information on how to successfully collect host and VM metrics.



SQL Diagnostic Manager displays virtual machine and host metrics as collected from the perspective of the physical host. In other words, these are the “true” physical metrics. Whereas, the metrics that SQL Diagnostic Manager displays for the OS on a virtualized machine are what Microsoft Windows calculates as using without regard to any overhead. For example, SQL Diagnostic Manager displays the OS CPU Usage at 35%, but metric displayed for the VM CPU Usage is 38%. This discrepancy results from the additional overhead within the hypervisor used to run the VM, and the VM is charged for the additional processing.

Link a monitored virtual SQL Server instance to the vCenter Server or Hyper-V host server

To link a monitored virtual SQL Server instance to the host server that manages your VMs, click **Tools > VM Configuration** to access the VM Configuration window. SQL Diagnostic Manager displays the current relationships between your available virtualization host servers and your monitored SQL Server instances. For more information about linking a monitored virtual SQL Server instance to a vCenter Server or Hyper-V host server, see [Configure your virtual machine connections](#).

VM data collection

The VM collector connects to your virtualization host server (vCenter Server or Hyper-V host server) during the scheduled refresh for each of your monitored virtual SQL Server instances. If the SQL Diagnostic Manager

Management Console is open to the Overview Dashboard or any Resources view for a specific instance, data is also collected based on the Server View Refresh setting in the Console Options page. In this situation, it is possible that the collector connects to the virtualization host server and virtual machines every few seconds.

You can adjust the scheduled refresh rate for a monitored virtual SQL Server instance by right-clicking the instance name, and then selecting **Properties**.

Differences in statistics

You may notice some difference in your statistics, such as SQL Diagnostic Manager displaying a higher OS CPU usage than your VM CPU usage. While the hypervisor manages resources efficiently, it is possible that the demand for physical resources is greater than what is available. For example, a guest OS on a virtual machine submits a batch of work to a CPU. However, all of the physical CPUs are committed to other work, so there is no physical CPU available on the host to process the work. In this situation, the VM waits for a CPU to become available to process its work. While it is waiting, the guest OS is unaware of the wait and assumes that it is simply taking longer for the CPU to process this batch of work, and the OS thinks it is using more CPU power than it is actually using.

Remember that SQL Diagnostic Manager uses the additional overhead of the hypervisor to calculate the VM metrics. In this situation, the hypervisor knows the VM is just waiting for a CPU, so it does not charge the VM for CPU processing power while it is waiting. The result is that the guest OS reports that it is using the additional processing power, however the hypervisor simply reports what is actually used. In this situation, your CPU usage metrics that are reported by Windows are higher than what is reported by VMware or Hyper-V for that VM.

Available metrics related to your VMs

A monitored virtual SQL Server instance appears in the Servers pane in the same way as a monitored physical SQL Server instances. Click the appropriate virtual instance, and SQL Diagnostic Manager displays the same dashboard panels as for a monitored physical SQL Server instance, but also include specific metrics for the VM and host server.

The following dashboard panels include VM- or host-related information:

- [CPU panel](#) - Includes VM and Host usage metrics. The information available in this panel comes from the Server Details and Resources Summary views.
- [Memory panel](#) - Includes the VM Memory Usage and Host Memory Usage charts. The information available in this panel comes from the Server Details and Resources Summary views.
- [Network panel](#) - Includes the VM Network Usage Throughput and Host Network Usage Throughput charts. The information available in this panel comes from the Server Details and Resources Summary views.
- [VM Disk Usage panel](#) - Includes the amount of data read from and written to disk for this VM during the sample interval. The information available in this panel comes from the Resources Disk view.
- [Host Disk Usage panel](#) - Includes the amount of data read from and written to disk for this host server during the sample interval. The information available in this panel comes from the Resources Disk view.
- [Virtualization panel](#) - Includes the Virtual Memory Usage and Virtual Disk Usage charts plus the CPU Ready Wait Time gauge. The information available in this panel comes from the Server Details view.

Use the available virtualization alerts and VM-related reports to get the most information regarding your monitored virtual SQL Server instances and host servers.

Available VM-related reports

- [Virtualization Statistics Report](#)
- [Virtualization Summary Report](#)

Available virtualization alerts

- [Host CPU Usage Alert](#)
- [Host Memory Swap Detected alert](#)
- [Host Memory Usage Alert](#)
- [Host Power State Alert](#)
- [VM CPU Ready Wait Time Alert](#)
- [VM CPU Usage Alert](#)
- [VM Host Server Change Alert](#)
- [VM Memory Swap Delay Detected Alert](#)
- [VM Memory Usage Alert](#)
- [VM Power State Alert](#)
- [VM Reclaimed Ballooned Memory Alert](#)
- [VM Resource Configuration Change Alert](#)
- [VM Resource Limits Detected Alert](#)


Hyper-V user requirements

In virtualized environment with Hyper-V, the process of how SQL Diagnostic Manager collects and presents metrics is slightly different from that of a VMware setup. To successfully collect Hyper-V host metrics, the user needs access to the WMI credentials on the host server. These credentials are established on the **Virtualization Host Configuration** window (Tools >VM Configuration >New or Tools >VM Configuration >View/Edit)

In addition, to successfully collect VM metrics, the user needs access to the WMI credentials on the VM. These credentials are established under on the **OS Metrics** tab of the **Monitored SQL Server Properties** window (right-click on the monitored instance name and select Properties).

Trial installation instructions

This procedure guides you through SQL Diagnostic Manager Trial installation.

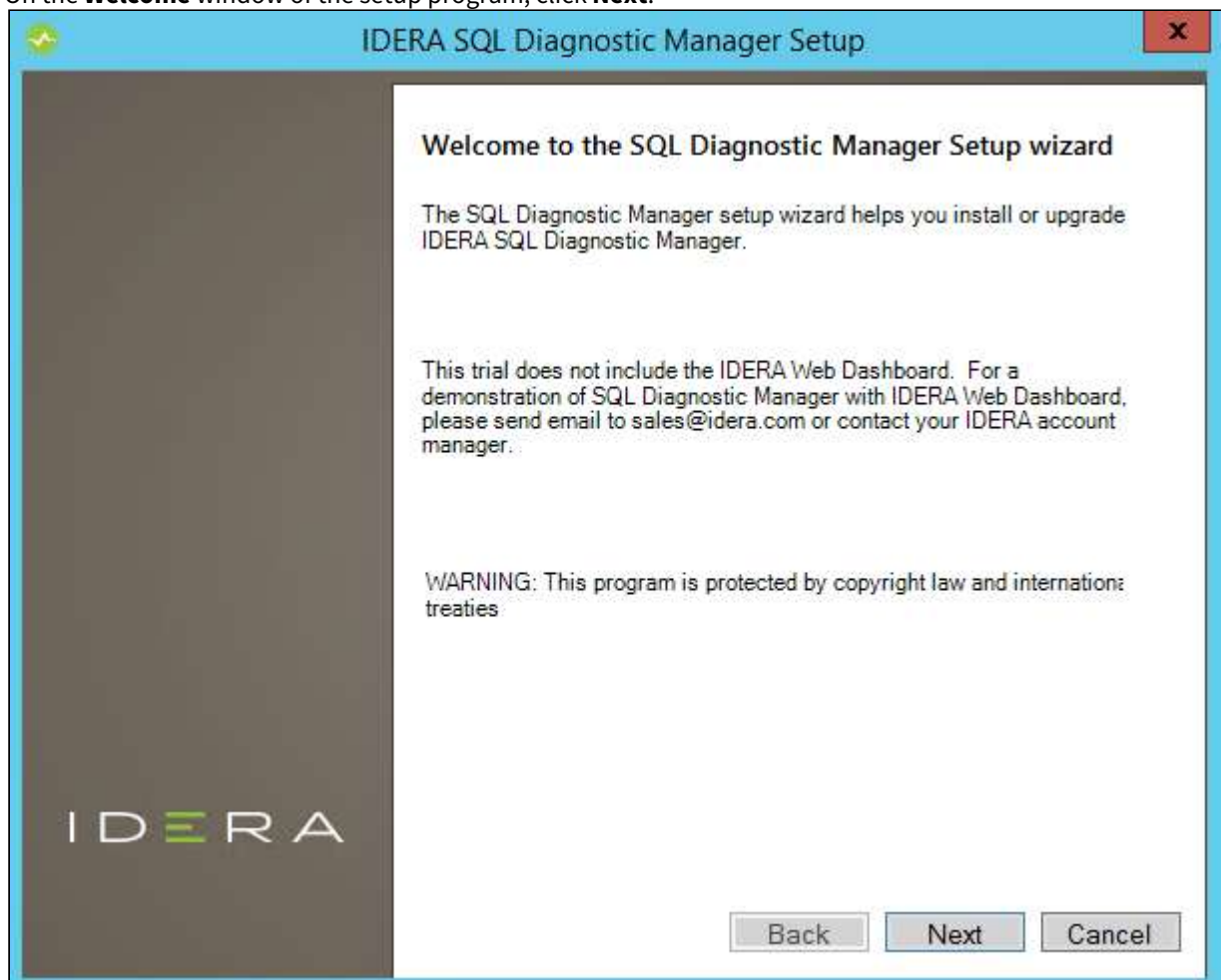
 The Trial does not include the IDERA Dashboard. If you wish to try it contact [IDERA Sales](#).

Start your Trial SQL Diagnostic Manager installation

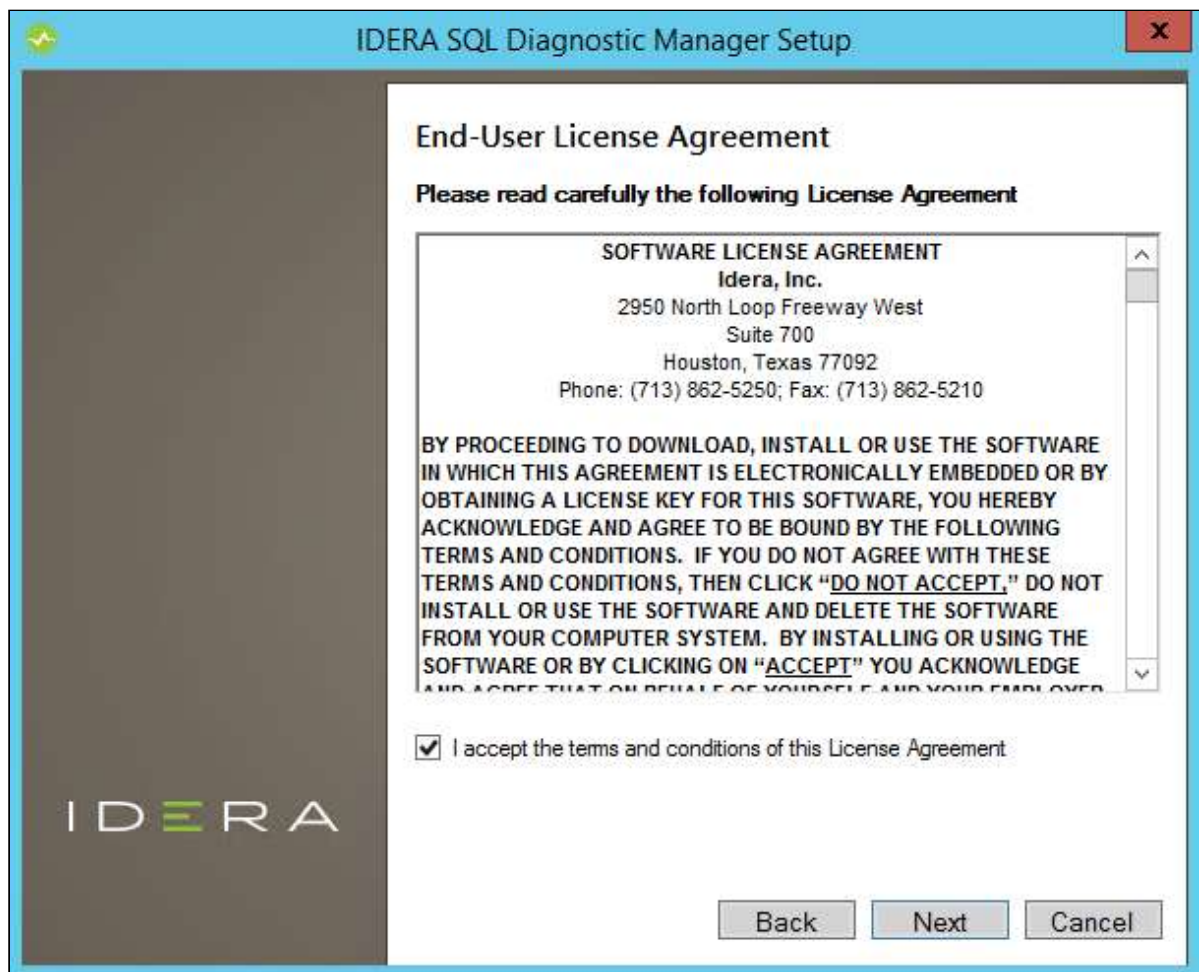
You can install SQL Diagnostic Manager on any computer that meets or exceeds the product requirements.

To start installing SQL Diagnostic Manager:

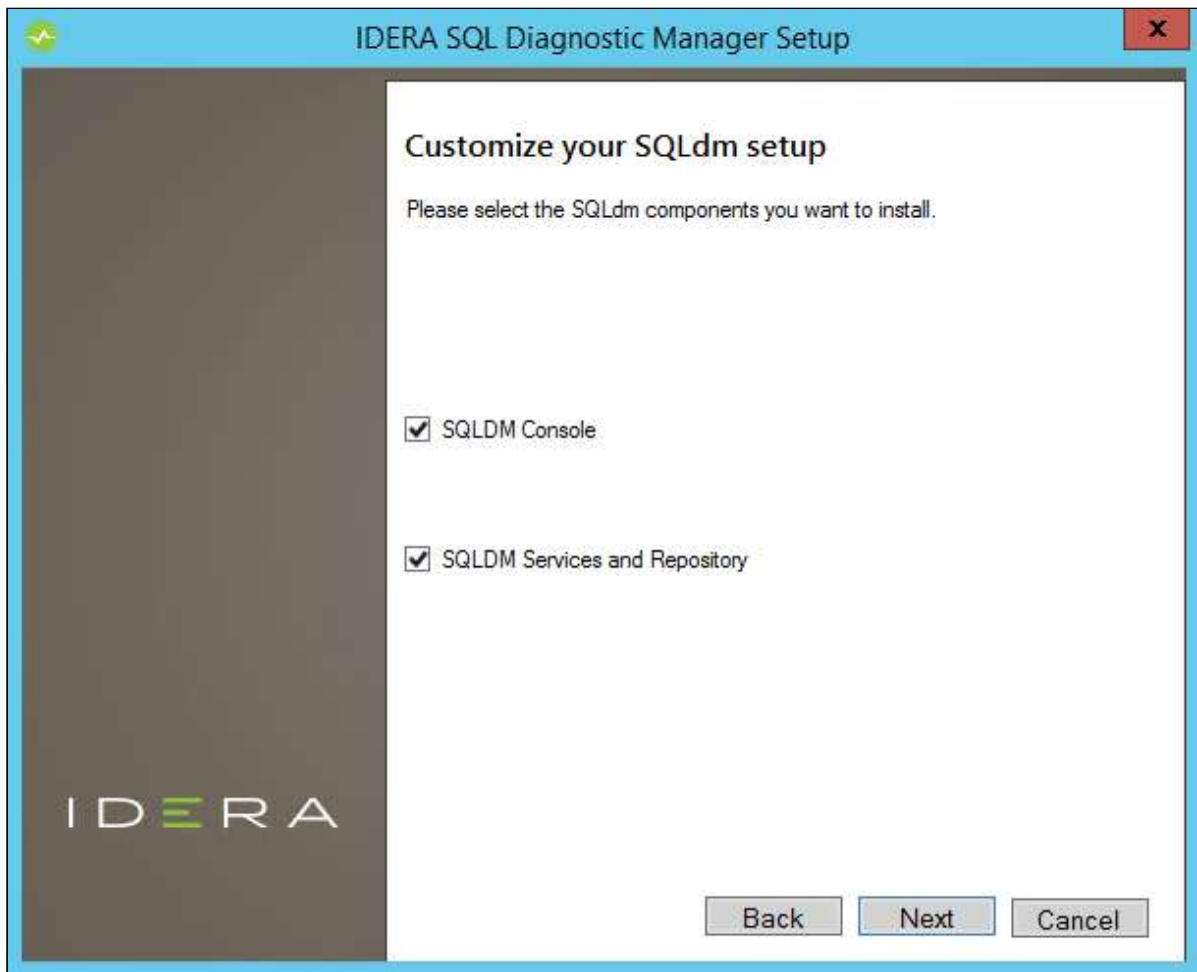
1. Run `Setup.exe` in the root of the Trial installation kit.
2. Click **Install SQL Diagnostic Manager** on the **Quick Start** window.
3. On the **Welcome** window of the setup program, click **Next**.



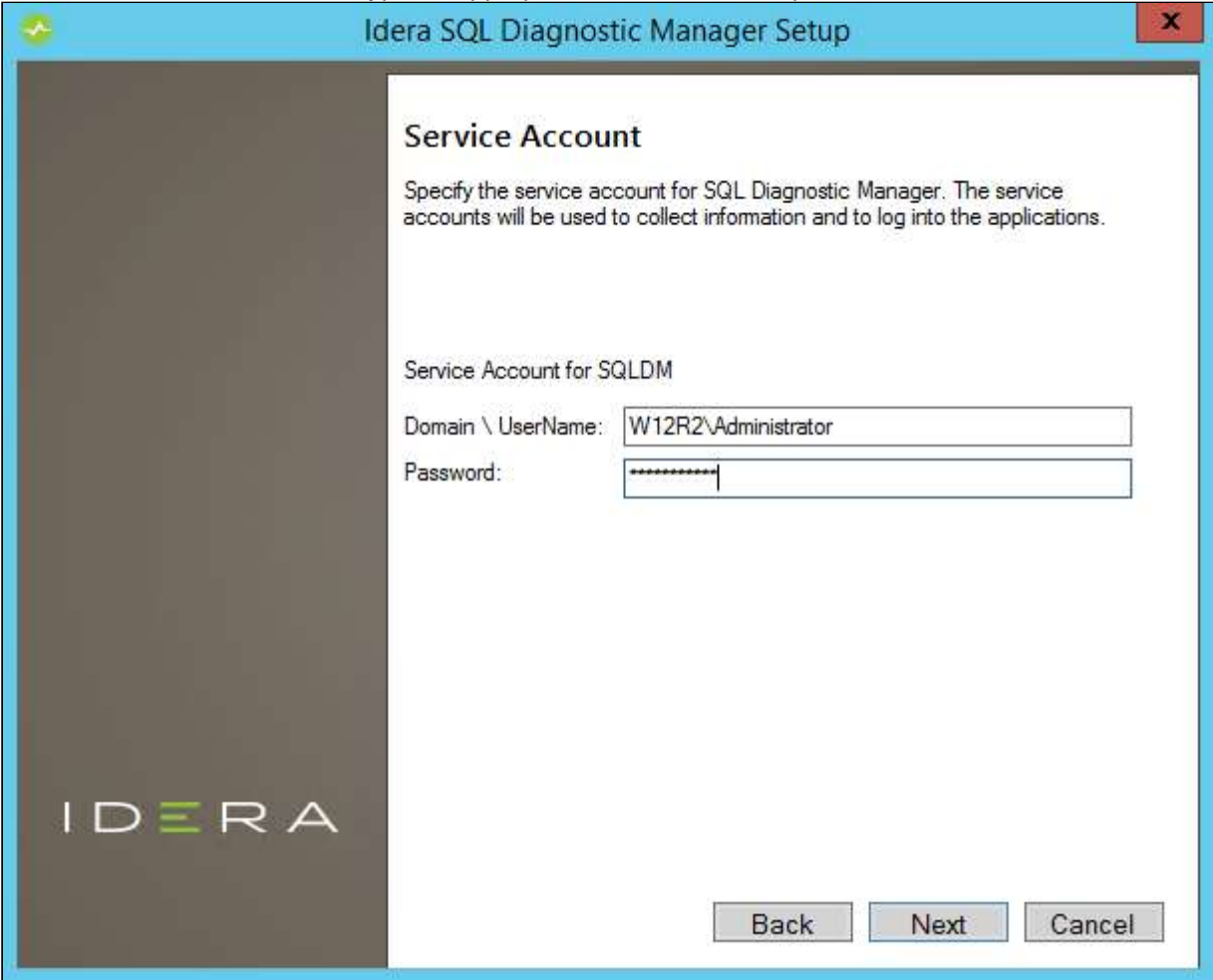
4. Review the license agreement. To accept this license agreement, click **I accept the terms and conditions of this License Agreement**, and click **Next**.



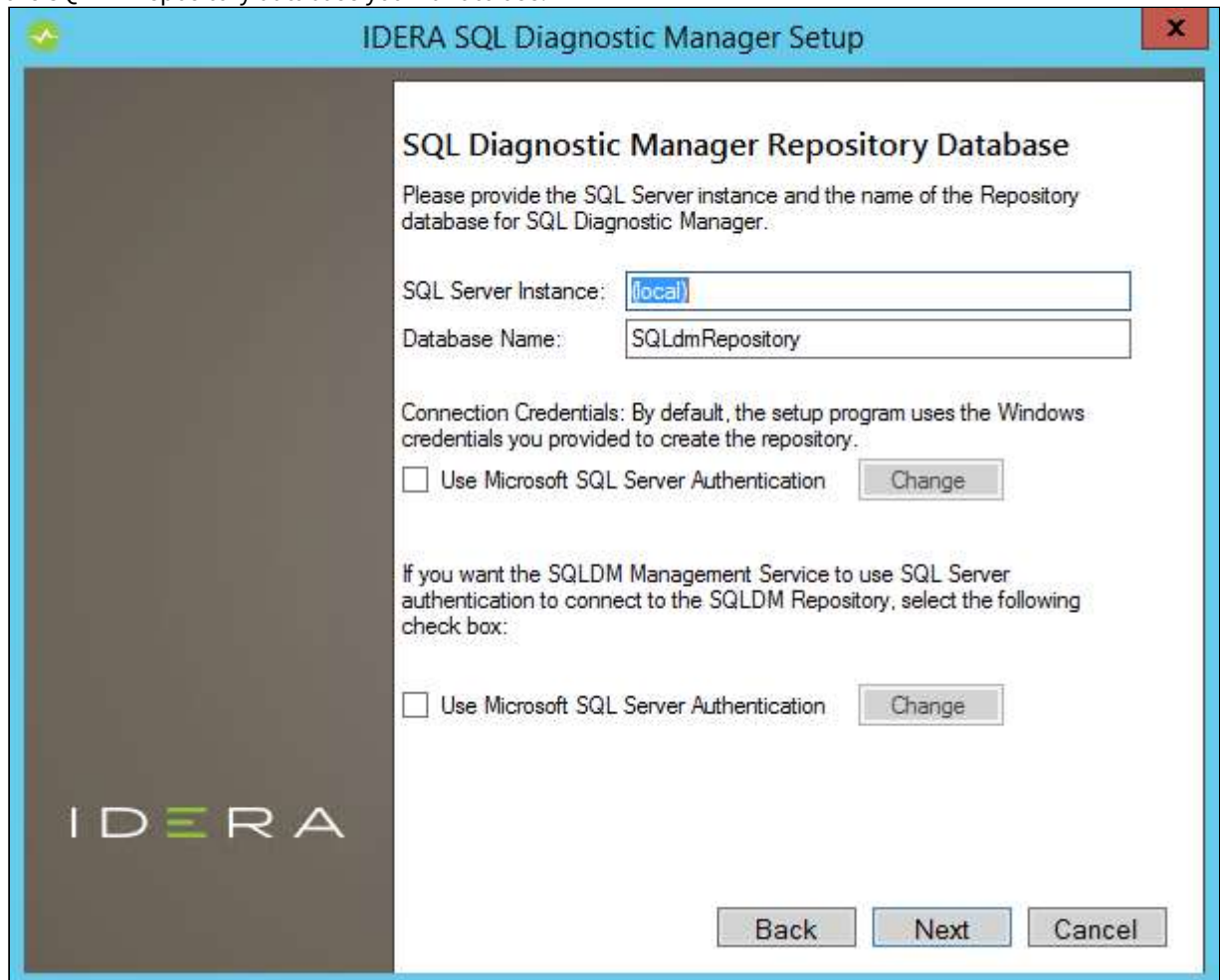
5. Choose the components you want to install by selecting the respective check boxes. By default, the setup program installs SQLDM Console, and SQLDM Repository and Services, allowing you to immediately begin monitoring your SQL Server environment, and click **Next**.



6. On the Service Account window, type the appropriate credentials in the provided fields. Click **Next**.



7. On the SQL Diagnostic Manager Repository Database screen, specify SQL Server instance and the name of the SQLDM Repository database you want to use.



The screenshot shows the 'IDERA SQL Diagnostic Manager Setup' window. The title bar is blue with the IDERA logo on the left and a close button on the right. The main content area has a dark grey sidebar on the left with the IDERA logo. The main panel is white and contains the following text and controls:

SQL Diagnostic Manager Repository Database

Please provide the SQL Server instance and the name of the Repository database for SQL Diagnostic Manager.

SQL Server Instance:

Database Name:

Connection Credentials: By default, the setup program uses the Windows credentials you provided to create the repository.

Use Microsoft SQL Server Authentication

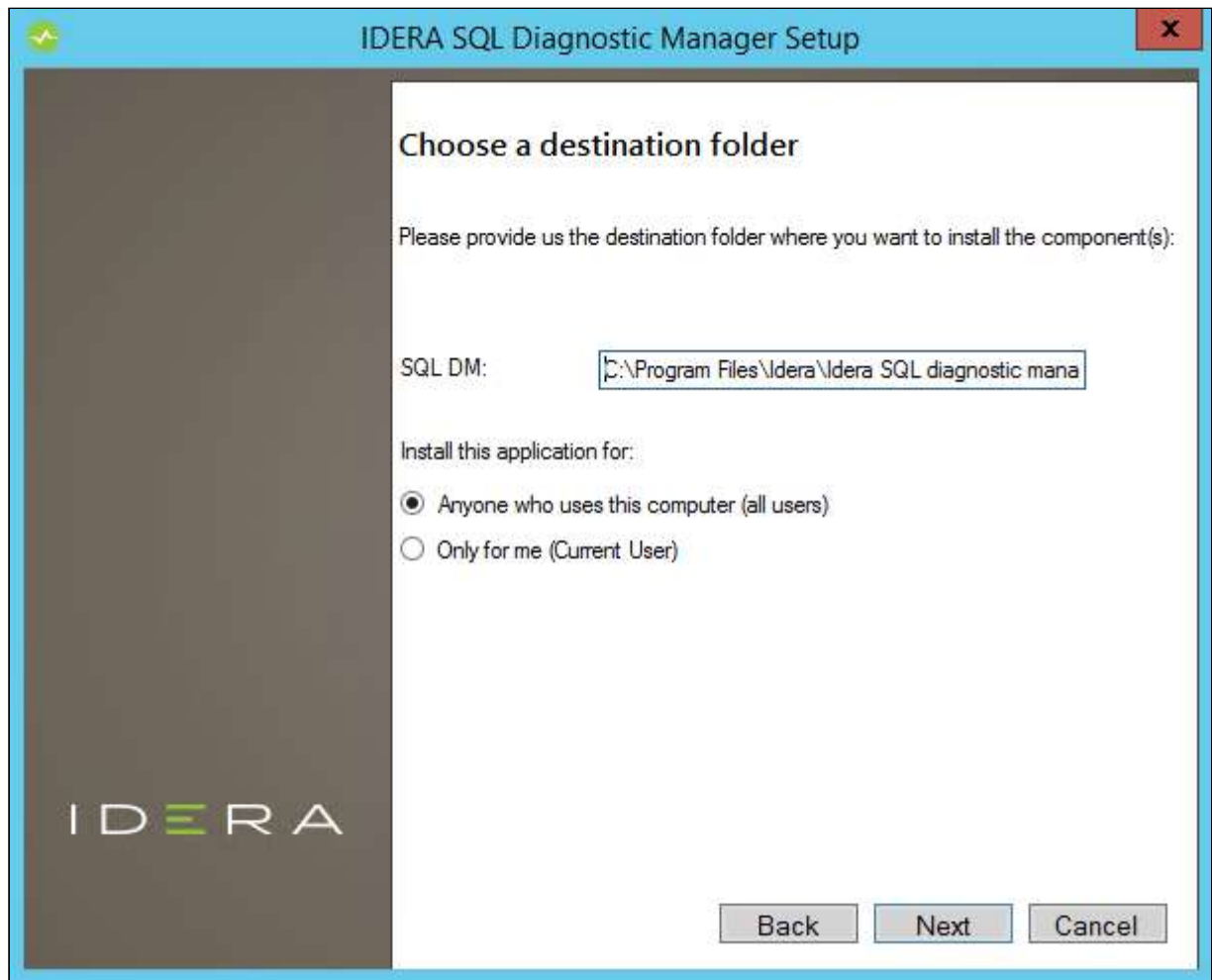
If you want the SQLDM Management Service to use SQL Server authentication to connect to the SQLDM Repository, select the following check box:

Use Microsoft SQL Server Authentication

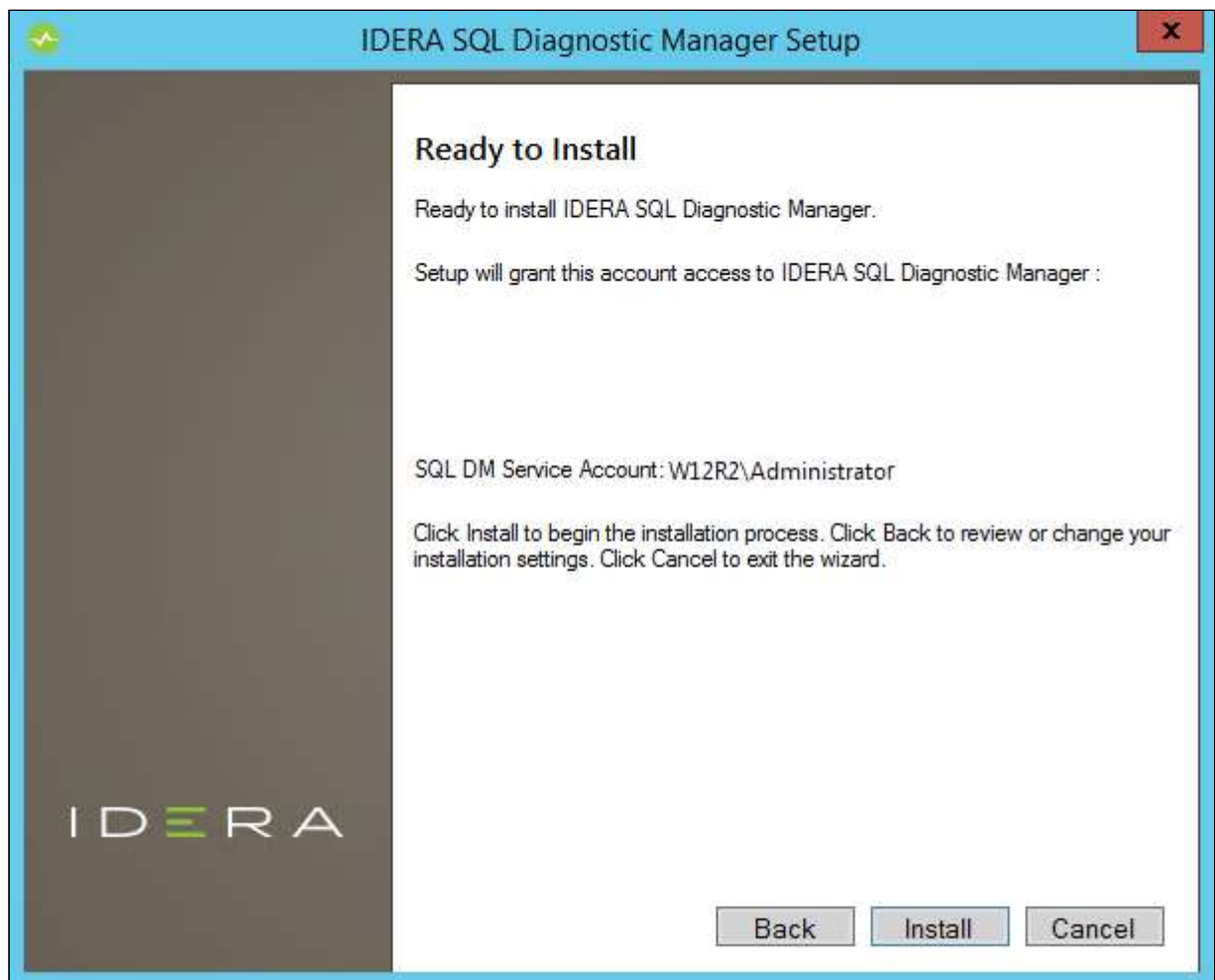
At the bottom right, there are three buttons:

! When your SQL Server environment uses SQL Server Authentication, select the respective check box, and click **Change** to enter SQL Server credentials. And, if you use Microsoft SQL Server Authentication, a new window displays where you can type the respective SQL Server login and password.

8. You can use the default install location or specify a different path. For your first install, we recommend using the default location. Click **Next** to continue with the installation.



9. Click **Install** to indicate that you are ready to complete your installation with the configurations you specified. Click **Back**, if you want to review or change your installation settings.



10. Click **Finish** to exit the IDERA SQL Diagnostic Manager Setup.

Configure your deployment

After initially installing SQL Diagnostic Manager, use the following configuration tasks to help you get the most out of SQL Diagnostic Manager:


- [Add your SQL Server instances](#)
- [Configure console options](#)
- [Connect to a SQL Diagnostic Manager Repository](#)
- [Set your Repository grooming schedule](#)
- [Use tags to organize your monitoring environment](#)
- [Use custom counters to track metrics](#)

Connect to the SQL Diagnostic Manager Repository

The first time you open the SQL Diagnostic Manager Management Console, SQL Diagnostic Manager displays the Connect to SQL Diagnostic Manager Repository window. Use this window to select the SQL Server instance, SQL Diagnostic Manager Repository database name, and enter the authentication details the SQL Diagnostic Manager Console uses to communicate with the SQL Diagnostic Manager Repository.

The SQL Diagnostic Manager Console requires Read Access on the SQL Diagnostic Manager Repository database. You can use either Windows Authentication or SQL Server Authentication for this connection.

Access the Connect to SQL Diagnostic Manager Repository window


You can open the Connect to SQL Diagnostic Manager Repository window from the SQL Diagnostic Manager Console, selecting  > **Connect to SQLDM Repository**. The [ability to connect to another Repository](#) is important in disaster recovery situations where you need to change the location of your SQL Diagnostic Manager Repository.

Add your SQL Server instances

The Add Servers wizard allows you to add the SQL Server instances you want to monitor. The Add Servers wizard also allows you to configure all the common options you use for monitoring SQL Server collection settings and [alerts](#), and apply an alert template.

- ✔ If you are adding several SQL Server instances and you want to configure the same alert thresholds on each instance, consider [setting up an alert template](#) before adding any of your SQL Server instances.

Access the Add Servers wizard


You can open the Add Servers wizard from the SQL Diagnostic Manager Console menu by selecting  > **Manage Servers**, and clicking **Add** in the Manage Servers window.

Other configurations that need attention when adding a virtualized SQL Server instance

If the monitored SQL Server instance you want to add resides on a virtual machine, add the instance using the Add Servers wizard, and then [access the VM configuration window](#) to associate the instance with the correct virtual machine and host.

- ✔ IDERA recommends that you register individual instances in the Add Servers wizard when looking to monitor availability replicas. Note that registering an Availability Group Listener may result in misleading data due to the failovers between availability replicas.

About alert templates

SQL Diagnostic Manager allows you to configure generic alert settings as a template that you can apply to servers and groups of servers in your organization. Select  > **Alert Configuration Template**, and complete the required fields to [configure an alert template](#).

Delete a SQL Server instance


You can easily delete a SQL Server instance at any time by right-clicking the appropriate instance, and selecting **Delete**. SQL Diagnostic Manager displays a confirmation message that asks, "Would you like to retain the data collected for the selected instance?" Click **Yes** and SQL Diagnostic Manager retains data for the instance and includes it in some reports, such as the Server Inventory report. Click **No** and SQL Diagnostic Manager deletes the SQL Server instance and all related data.

Configure authentication

The Configure Authentication window of the Add Servers wizard allows you to select the authentication credentials SQL Diagnostic Manager uses when collecting performance and statistics information from your SQL Server instances. You can also set encryption to secure the connection.

After selecting a connection type and encryption, click **Next** to continue.

Access the Add Servers wizard

You can open the Add Servers wizard from the SQL Diagnostic Manager Console menu by selecting  > **Manage Servers**, and clicking **Add** in the Manage Servers window.

Select a connection type

Windows Authentication is the recommended collection type as SQL Diagnostic Manager uses a number of SQL trace statements that require a trusted connection. When you select and use **Windows Authentication**, the account the SQLdm Collection Service is running as is used to connect to the SQL Server instance.

Select encryption

SQL Diagnostic Manager allows you to designate encryption methods used to encrypt data between the SQLdm Collection Service and the monitored SQL Server instance. The options include SSL (Secure Sockets Layer) or SSL with Trust Service Certificate.

Encrypt Connection (SSL)

The **Encrypt Connection (SSL) option** adds a flag in the connection properties that triggers SSL use when the SQLdm Collection Service connects to a monitored server that uses SSL to encrypt the data transfer. For this option to work correctly, you must configure the monitored SQL Server instance to use SSL encryption.

Trust Server Certificate (Bypass Certificate Validation)

The **Trust Server Certificate (bypass certificate validation)** option is available only after you select **Encrypt Connection (SSL)**. This option allows you to skip certificate validation when a SQL Server instance establishes a connection. If SSL on the monitored SQL Server instance is not configured to use a certificate that the collection service trusts, then the connection is rejected unless you select the Trust Server Certificate option.


Select instances to monitor

The Select Servers to Monitor window of the Add Servers wizard allows you to select one or more of the SQL Server instances you want to monitor using SQL Diagnostic Manager. If the name of the SQL Server instance you want to monitor does not appear in the list, type the instance name in the space provided in the Available Servers area. You can also designate specific ports for your SQL Server instances by entering a comma between the instance name and port number, for example domain\instance name,port.

The **Available Licenses** field shows you the number of instances your current license permits SQL Diagnostic Manager to monitor.

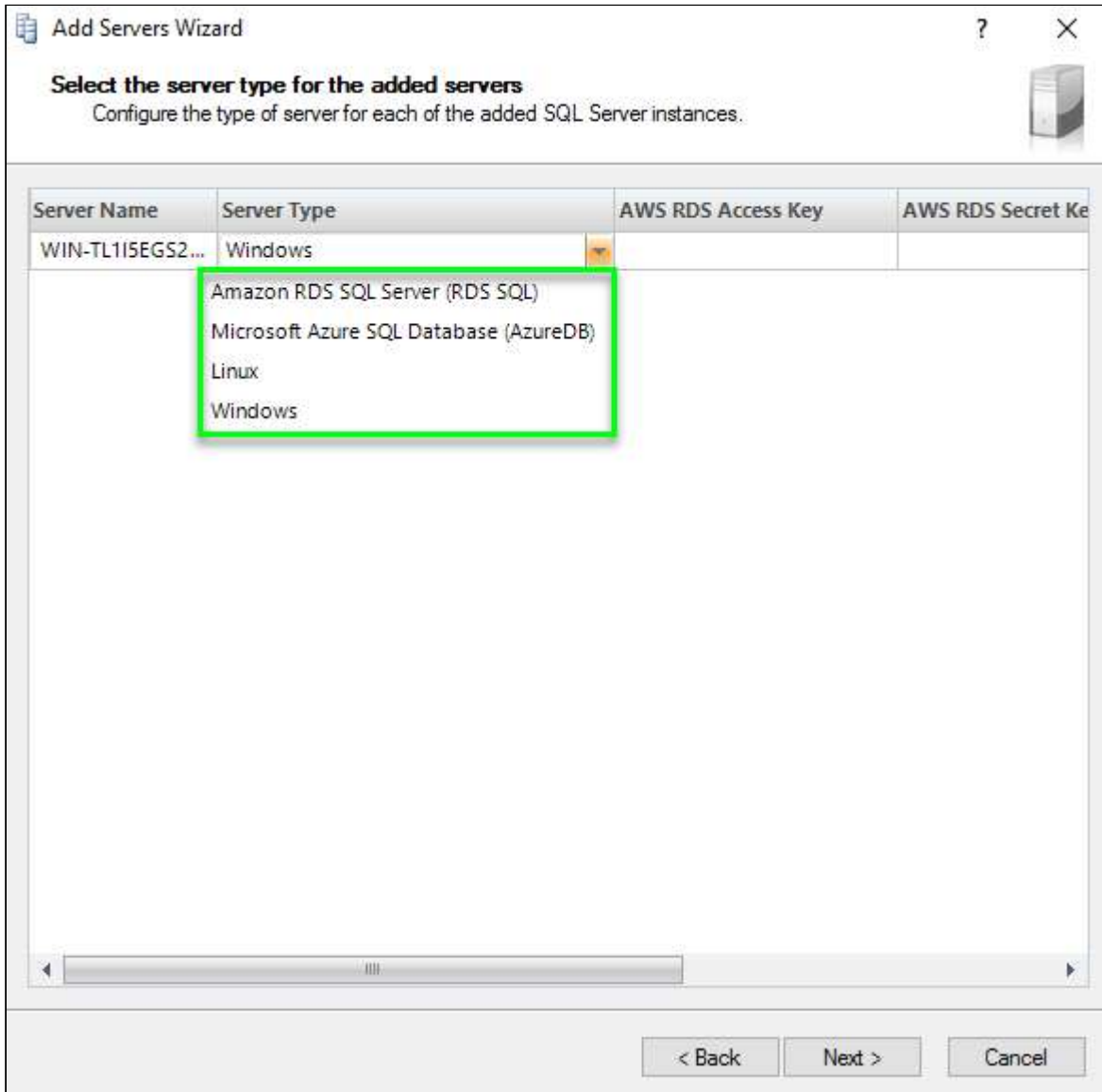
- ✔ The Select Servers to Monitor window lists all the SQL Server instances provided by your domain controller. This list can include unsupported SQL Server instances. Be sure to select only [SQL Server instances that are supported by SQL Diagnostic Manager](#).

Access the Add Servers wizard

You can open the Add Servers wizard from the SQL Diagnostic Manager Console menu by selecting  > **Manage Servers**, and clicking **Add** in the Manage Servers window.

Select cloud or Linux provider


Configure the type of server (Amazon RDS SQL Server, Azure SQL Database, Linux, or Windows) for each of the added SQL Server instances.




The select cloud providers for the added servers window allows you to choose a cloud provider for your monitored SQL Server instances that are hosted on the cloud (PaaS). SQL Diagnostic Manager supports Amazon RDS and Microsoft SQL Azure (SQL Azure Database, elastic pool, and managed instances).

i Azure SQL and Amazon RDS permissions

Review the [Minimum permissions to monitor an Amazon RDS instance](#) and the [Minimum Permissions for Azure SQL](#) to create a SQL Server user on the server to monitor a cloud instance.


 In addition, SQLDM supports monitoring of images hosted on virtualized or cloud IaaS environments such as SQL Server on Azure virtual machines and Amazon EC2. Monitoring a virtualized instance requires no cloud-specific configuration. Select "Not a cloud instance" for a virtualized or cloud IaaS instance.


Access the Add Servers wizard


You can open the Add Servers wizard from the SQL Diagnostic Manager Console menu by selecting  > **Manage Servers**, and clicking **Add** in the Manage Servers window.

Metric availability for instances hosted on the cloud

SQL Diagnostic Manager allows you to view a wide range of information collected for your PaaS monitored instances hosted on Amazon RDS and Microsoft SQL Azure (Azure SQL Database, elastic pool and managed instances).

 When monitoring Amazon RDS servers, SQL Diagnostic Manager cannot monitor or alert on SQL Agent jobs.

 In addition, SQLDM supports monitoring of images hosted on virtualized or cloud IaaS environments such as SQL Server on Azure virtual machines and Amazon EC2. Monitoring a virtualized instance requires no cloud-specific configuration.

 SQL Diagnostic Manager has new DBaaS metrics and alerts, for more information review the [Metric Alerts](#).

To have more information on the metrics data that are available for cloud instances, select your preferred option:

- [Available metrics for Amazon RDS and Microsoft SQL Azure in SQL Diagnostic Manager.](#)
- [Unavailable metrics for Amazon RDS and Microsoft SQL Azure in SQL Diagnostic Manager.](#)

Available metrics for Amazon RDS and Microsoft SQL Azure in SQL Diagnostic Manager

For monitored SQL Server instances hosted on Amazon RDS

SQL Diagnostic Manager Area	Collected Metrics by Amazon RDS
Overview → Details	Batches
	Batches Per Second
	Blocked Sessions
	Buffer Cache Hit Ratio
	Buffer Cache Size
	Checkpoint Writes
	Checkpoint Writes Per Second
	Client Computers
	Connections
	Connections Per Second
	Data File Count
	Data File Space Allocated

Database Count
Full Scans
Full Scans Per Second
Lazy Writer Writes
Lazy Writer Writes Per Second
Lead Blockers
Lock Waits
Lock Waits Per Second
Log File Count
Log File Space Allocated
Log File Space Used
Log Flushes
Log Flushes Per Second
Maintenance Mode
OLE Automation Status
Open Transactions
OS Metrics Availability
Packet Errors
Packet Errors Per Second
Packets Received
Packets Received Per Second
Packets Sent
Packets Sent Per Second
Page Life Expectancy
Page Lookups

Page Lookups Per Second
Page Reads
Page Reads Per Second
Page Splits
Page Splits Per Second
Page Writes
Page Writes Per Second
Procedure Cache Hit Ratio
Read Ahead Pages
Read Ahead Pages Per Seconds
Read Write Errors
Response Time
Running Since
Server Version
Sql Compilations
Sql Compilations Per Second
Sql Memory Allocated
Sql Memory Used
Sql Recompilations
Sql Recompilations Per Second
System Processes
System Processes Consuming CPU
User Connections Percent
User Processes
User Processes Consuming CPU

	WMI Service Status
	Workfiles Created
	Workfiles Created Per Second
	Worktables Created
	Worktables Created Per Second
Sessions → Summary	Response Time
	Sessions
	Blocked Sessions
	Lock Statistics → Average Wait Time
	Lock Statistics → Deadlocks
	Lock Statistics → Requests
	Lock Statistics → Timeouts
	Lock Statistics → Waits
	Lock Statistics → Wait Time
Sessions → Details	Connection
	User Session
	Status
	Host
	Net Library
	Network Address
	User
	Application
	Database
	Transaction Isolation Level
	Usage

Login Time
Last Activity
Physical I/O
CPU (ms)
Memory (kb)
Open Transactions
Lock Information
Wait Time (ms)
Blocking
Blocking Count
Tempdb Usage
Version Store Elapsed Seconds
Space Used in kb (Alloc / Dealloc)
Session User Space
Task User Space
Session Internal Space
Task Internal Space
Last Command

For monitored SQL Server instances hosted on Microsoft SQL Azure

SQL Diagnostic Manager Area	Collected Metrics by Microsoft SQL Azure
Overview → Details	Response Time
Sessions → Summary	Response Time
	Sessions
	Blocked Sessions
Sessions → Details	Connection
	User Session

Status
Host
User
Execution Context
Application
Database
Transaction Isolation Level
Usage
Login Time
Last Activity
Physical I/O
CPU (ms)
Memory (kb)
Lock Information
Wait Time (ms)
Resource
Blocking
Blocking Count
Tempdb Usage
Version Store Elapsed Seconds
Space Used in kb (Alloc / Dealloc)
Session User Space
Task User Space
Session Internal Space
Task Internal Space

Last Command

Unavailable metrics for Amazon RDS and Microsoft SQL Azure in SQL Diagnostic Manager

SQL Diagnostic Manager allows you to view a wide range of information collected for your monitored SQL Server instances hosted on Amazon RDS and Microsoft SQL Azure. However, the following information and metric data are not available yet:

For monitored SQL Server instances hosted on Amazon RDS

SQL Diagnostic Manager area	Information and metric data not collected by Amazon RDS
Overview → Details	<ul style="list-style-type: none"> Available Memory Data File Space used Disk Queue Length Disk Time (Percent) OS CPU Privileged Activity OS CPU User Time OS Total Processor Activity Pages Per Second Physical Memory Procedure Cache Size Processor Queue Length Sql CPU Activity Sql Memory Used Percent Used Memory (Percent)
Sessions → Details	<ul style="list-style-type: none"> Connection Execution Context Usage

	Lock Information
	Lock Information
	Wait Type
	Resource

For monitored SQL Server instances hosted on Microsoft SQL Azure

SQL Diagnostic Manager area	Information and metric data not collected by Microsoft SQL Azure
------------------------------------	---

Overview → Details

Available Memory

Batches

Batches Per Second
Blocked Sessions
Buffer Cache Hit Ratio
Buffer Cache Size
Checkpoint Writes
Checkpoint Writes Per Second
Client Computers
Connections
Connections Per Second
Data File Count
Data File Space Allocated
Data File Space used
Database Count
Disk Queue Length
Disk Time (Percent)
Full Scans
Full Scans Per Second
Lazy Writer Writes
Lazy Writer Writes Per Second
Lead Blockers

Lock Waits
Lock Waits Per Second
Log File Count
Log File Space Allocated
Log File Space Used
Log Flushes
Log Flushes Per Second
Maintenance Mode
OLE Automation Status
Open Transactions
OS CPU Privileged Activity
OS CPU User Time
OS Metrics Availability
OS Total Processor Activity
Packet Errors
Packet Errors Per Second
Packets Received
Packets Received Per Second
Packets Sent
Packets Sent Per Second


Page Life Expectancy
Page Lookups Per Second
Page Reads
Page Reads Per Second
Page Splits
Page Splits Per Second
Page Writes
Page Writes Per Second
Pages Per Second
Physical Memory
Procedure Cache Hit ratio
Procedure Cache Size
Processor Queue Length
Read Ahead Pages
Read Ahead Pages Per Second
Read Write Errors
Running Since
Server Version
Sql Compilations
Sql Compilations Per Second

	Sql CPU Activity
	Sql Memory Allocated
	Sql Memory Used
	Sql Memory Used Percent
	Sql Recompilations
	Sql Recompilations Per Second
	System Processes
	System processes Consuming CPU
	Used Memory(Percent)
	User Connections Percent
	User Processes
	User Processes Consuming CPU
	WMI Service Status
	Workfiles Created
	Workfiles Created Per Second
	Worktables Created
	Worktables Created Per Second
Sessions → Summary	Lock Statistics → Average Wait Time
	Lock Statistics → Deadlocks
	Lock Statisttics → Requests

	Lock Statistics → Timeouts
	Lock Statistics → Waits
	Lock Statistics → Wait Time
Sessions → Details	Connection
	Net library
	Network Address
	Usage
	Open Transaction
	Lock Information
	Wait Type

User Permissions for Azure

The minimum permission required for Azure SQL is *Server admin*.


 For more information about the characteristics of the Server admin and Azure Active Directory admin accounts, review [Controlling and granting database access to SQL Database and SQL Data Warehouse](#).

Configure data collection

The Configure SQL Diagnostic Manager Collection window of the All Servers wizard allows you to set the interval at which you want to collect diagnostic data and raise alerts, and whether you want to enable query monitoring.


The collection interval you specify refers to the time SQL Diagnostic Manager waits to collect statistical information on your SQL Server instance. This data includes the resource, session, database and tables, query, and service information.

The Query Monitor is a standard SQL Server trace that collects events occurring on your SQL Server instance over a period of time.

 Enabling the Query Monitor can cause SQL Server performance degradation while it runs. Enable the Query Monitor only when diagnosing specific query issues.

After making any changes to the collection interval or enabling the Query Monitor, click **Next** to continue.

Access the Add Servers wizard


You can open the Add Servers wizard from the SQL Diagnostic Manager Console menu by selecting  > **Manage Servers**, and clicking **Add** in the Manage Servers window.

Select a collection interval

Choosing the appropriate interval depends on your specific monitoring needs. When your SQL Server instance performs poorly, you can decrease the collection interval so that you can track changes more often. When monitoring a SQL Server instance that has a very heavy client load and a large number of databases, you can increase the time period between collections to reduce the possibility of the frequent collections causing a performance impact on your clients.

Monitoring your environment by collecting query activities

When you experience query timeouts or other performance issues, you can collect query-related activities through the Query Monitor. Note that enabling these options can degrade performance on the monitored SQL Server instance because of the additional data collection and storage. To help alleviate degradation, SQL Diagnostic Manager allows you to select only those types of queries you want to monitor such as batches, statements, procedures, and triggers. Additional threshold fields allow you to set the levels at which you consider a query as performing poorly.

 It is important to remember that once you complete your problem diagnosis, you can disable the Query Monitor on the Monitored SQL Server Properties window by selecting the Query Monitor tab and then clearing the **Enable the Query Monitor** check box.

Using advanced Query Monitor options

Query Monitor includes advanced options that allow you to filter applications, databases, and SQL text from collection. Use semicolons (;) to separate multiple entries. You can also use the percent character (%) as a wildcard.

You can access this window by clicking the **Advanced** button in the Query Monitor tab of the monitored SQL Server Properties window.

Configure OS metric collection

The Configure OS Metric Collection window of the Add Servers wizard lets you select which type of operating system metrics collection you want to use or whether you do not want to collect any OS metrics from a SQL Server instance:


- Do not collect Operating System data
- Collect Operating System data using OLE Automation
- Collect Operating System data using direct WMI (selected by default when adding a new SQL Server instance)

SQL Diagnostic Manager uses OLE automation or a direct connection to the Windows Management Interface (WMI) when collecting OS metrics and service status data on clustered SQL Servers. For additional information about OS metrics collection, see [Configure OS metrics monitoring](#).

✔ SQL Diagnostic Manager retains your previous settings upon upgrade if OLE automation is the default for your monitored SQL Server instance.

After selecting your method for OS metrics collection, click **Next** to continue.

Access the Add Servers wizard

You can open the Add Servers wizard from the SQL Diagnostic Manager Console menu by selecting  > **Manage Servers**, and clicking **Add** in the Manage Servers window.

Disabling OS metric collection

SQL Diagnostic Manager does not require OS metrics collection. However, if you disable this feature, SQL Diagnostic Manager ceases to collect OS metrics and does not raise any previously-associated alerts. You can easily enable OS metric collection at any time if you decide that you want to continue collecting these metrics.

Permissions for direct WMI collection

It is preferred that the WMI user is a local administrator on the monitored SQL Server instance. However, if you do not want to grant administrator access, use the following steps to configure remote WMI access in Microsoft Windows:

1. Add the user account to the Performance Log Users and Distributed COM users groups.
2. Grant privileges to WMI.

You also may need to add the WMI user account to the following policies:

- Act as part of the operating system
- Log on as a batch job
- Log on as a service
- Replace a process level token

For more information about using a direct WMI connection, see the Microsoft document, [Securing a Remote WMI Connection](#).

Using direct WMI when the collection service is on a different domain than the SQL Server

In environments where cross-domain trust security is enforced, data from asynchronous collection is not returned from an untrusted domain to a trusted domain. As a result, the collection service does not receive the data.

If you select **Collect Operating System data using direct WMI** and the machine used for your collection service is in a different domain than your monitored SQL Server instance, you must make some configuration changes on your collection service machine. If you do not make these changes, change your OS metrics collection to **Collect Operating System data using OLE automation** if you want OS and disk statistics plus the auto discovery of mount points.

Select an alert template and tags


The Select Alert Template and Tags window lets you assign an alert template and tags while you add an instance. Alert templates allow you to configure generic alert settings to apply to your servers or groups of servers. Tags are labels that help you organize server instances into meaningful groups.

To apply an alert template, select the appropriate template from the list. If you do not have an existing custom alert template, you can [create an alert template](#) after you add your SQL Server instance, and then apply the template to your server. You may select one or more existing tags from the list, or create new tags by clicking **Add Tag**.

If you create an instance before creating any custom alert templates, SQL Diagnostic Manager applies the default alert template automatically. You can then create an alert template and apply it to the server once the new template is available.

After selecting your baseline collection, click **Next** to continue.

Access the Add Servers wizard

You can open the Add Servers wizard from the SQL Diagnostic Manager Console menu by selecting  > **Manage Servers**, and clicking **Add** in the Manage Servers window.

Using tags

Tags are customizable to suit your needs. For example, you may want to view only server instances used by a particular department (like Accounting), in a specific location, or by some other method (like SQL Server version).

You can assign multiple tags to each instance so that the instance appears in several different groups.

Configure your virtual machine connections

The VM Configuration window allows you to associate your monitored virtual SQL Server instances with the virtualization host server that manages your VMs and the VM on which the monitored SQL Server instance is running.

Virtual machines appearing in the list are based on the permissions for the user account typed into the **Virtualization Host Configuration** dialog box. To view a specific VM, you must have the View permission on all parent folders.

- ✔ Remember to renew your link after changing the name of a virtual machine. Once the name change is complete, simply open the VM Configuration window and wait until after SQL Diagnostic Manager is done contacting the vCenter server or Hyper-V Host server to get a list of your VM names. Note that your updated VM name already appears in the list. Click **OK** to save the new name to the list of VMs. If you do not update the name, this action occurs again the next time you access the VM Configuration window.

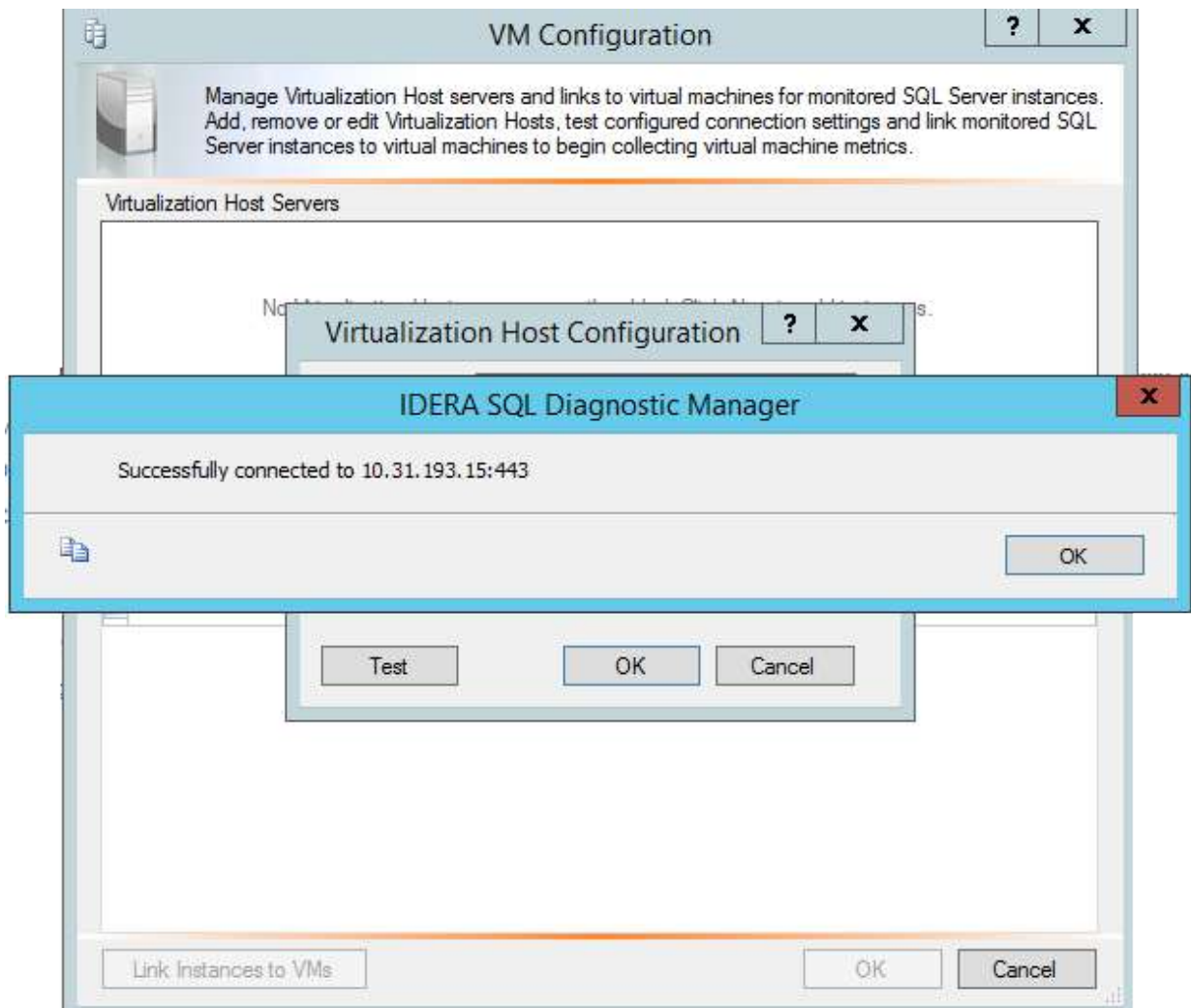
Access the VM Configuration window

To open the VM Configuration window, select **Tools > VM Configuration**, or click **VM Configuration** from the Manage Servers window. You can also access this window from the Virtualization tab of the Server Properties window by clicking the **VM Configuration** button.

Add a host server

SQL Diagnostic Manager allows you to add a new vCenter Server through the VM Configuration window, follow these steps to add one:

1. Click **New**.
2. In the Virtualization Host Configuration window fill in the blanks as follows:
 - a. **Server:** The vCenter server that has to be written as the **IP or vCenter Server name:443** (Default port #443).
 - b. **Name:** Use a friendly name to distinguish your server from other vCenters.
 - c. **User and password:** Use the domain account that is running the SQLDM Services that has read access to vCenter.
3. Click **Test** to test the configuration data.
4. Click **OK** to add the vCenter Server to the list of Virtualization Host Servers.



Hyper-V and metrics collection

See [Hyper-V user requirements](#) for information on how to successfully collect Hyper-V host and VM metrics.

Verify your connection to the host server

You can test the connection to your listed host servers by selecting the appropriate server from the list, and then clicking **Test**.

SQL Diagnostic Manager displays an icon with your current connection status. If the connection fails, click **View/Edit** and verify your connection details. Make the necessary changes to access your host server, click **Test**, and then click **OK** to save your changes after a successful test.

Associate a monitored SQL Server instance to a VM

Once you add your SQL Server instances to SQL Diagnostic Manager for monitoring, you must associate them with their host server and virtual machine.

To associate a monitored SQL Server instance to a virtual machine:


1. Verify that the instance you want to associate to the host appears in the list.
2. Click the **Virtualization** column for the appropriate virtual instance, and then select the associated host server from the drop-down list.
3. Click the **Virtual Machine** column, and then select the name of the virtual machine where this instance resides. SQL Diagnostic Manager populates this drop-down list based on the servers on the selected host server.
4. Click **Link Instances to VMs**.
5. Click **OK** to save your changes and close this window.

Configure your VM host server connection

The vCenter Connection Configuration window allows you to add or update connection details for your virtual machine host servers. Complete the access information for the server, and then click **Test** to make sure you typed the correct access credentials. Click **OK** after a successful test.

Occasionally, a connection test fails. Make sure you typed accurate information in the appropriate fields, and then click **Test**. If the test fails again, there may be an issue with the server to which you are trying to connect.

Access the vCenter Connection Configuration window


To access the vCenter Connection Configuration window, first access the VM Configuration window by selecting  **> VM Configuration**, or by clicking **VM Configuration** from the Manage Servers window. You can also access this window from the Virtualization tab of the Server Properties window. Once in the VM Configuration window, click **New**.

vCenter Server port requirements

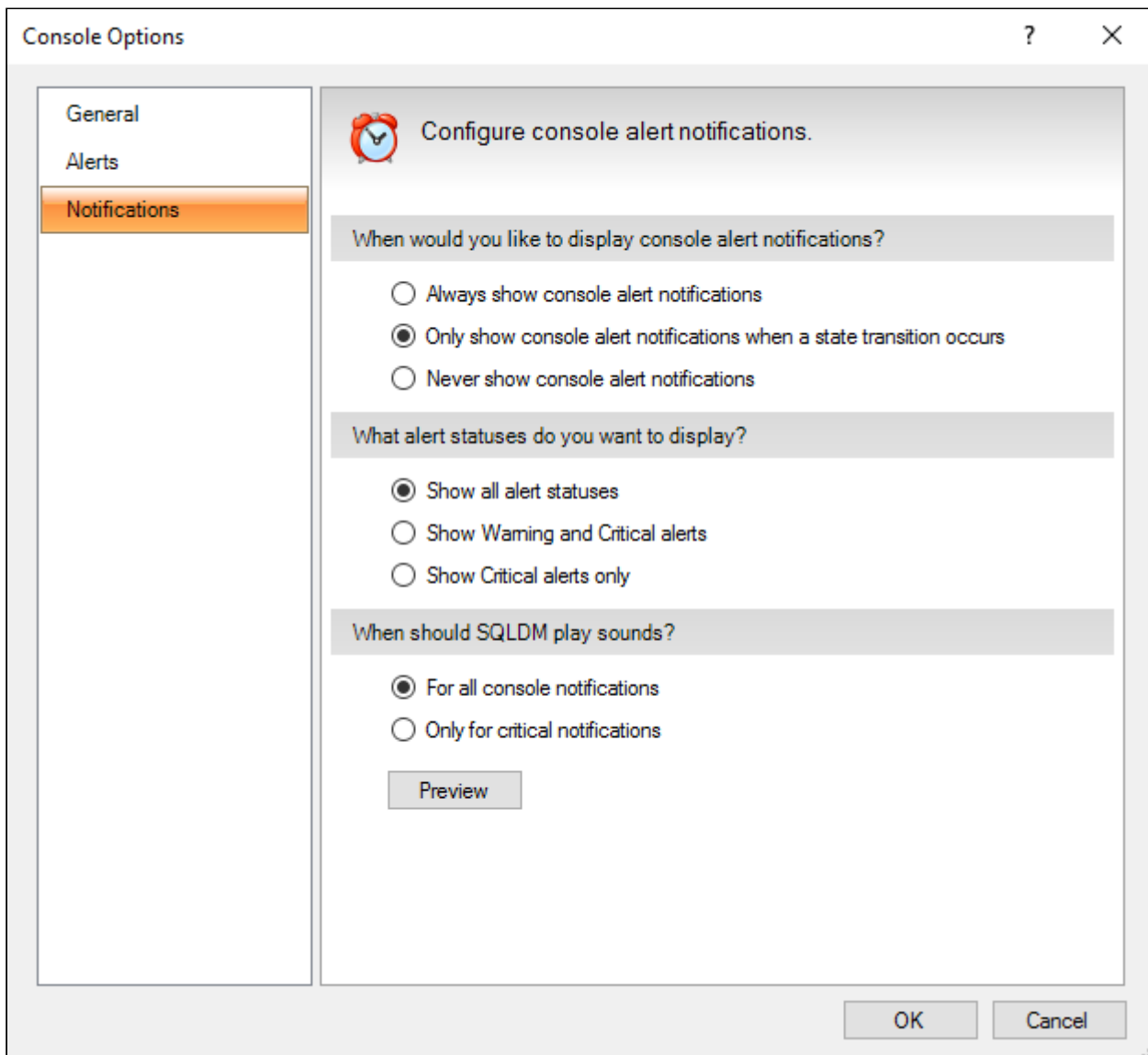
The vCenter Server requires that you open the ports included in the following table. For the most current list of ports, see the [VMware Knowledge Base article regarding network port requirements](#).

Port	Description
80	Required for direct HTTP connections. Note that this port redirects requests to HTTPS Port 443.
389	Active Directory Services for the vCenter Server group.
443	Listens for connection from the vSphere Client, vSphere Web Access Client, and other SDL clients.
636	SSL port of the local instance for vCenter Linked Mode.
902 (UDP)	Used to send data to managed hosts.
902/903	Used by the vSphere Client to display virtual machine consoles.
5989 (TCP)	CIM transaction traffic (Hardware Status).
8080	vCenter Management Webservices HTTP.
8443	Secure connection for vCenter Management Webservices HTTPS.
60099	Used to stream inventory object changes to SDK clients. Firewall rules for this port on the vCenter Server can be set to block all, except from and to localhosts if the clients are installed on the same host as the vCenter Server service.

Configure console options

You can specify a number of console options in the Console Options window that affect overall product behavior. To access the Console Options window, select  > **Console Options** from the SQL Diagnostic Manager Console toolbar.

The following tabs are available on the Console Options window:



General

The [General tab](#) allows you to set how often the console refreshes data, hide the Console when minimized, select how SQL Diagnostic Manager manages chart data, choose what colors you want SQL Diagnostic Manager to use, and enable the Baseline view.

Alerts

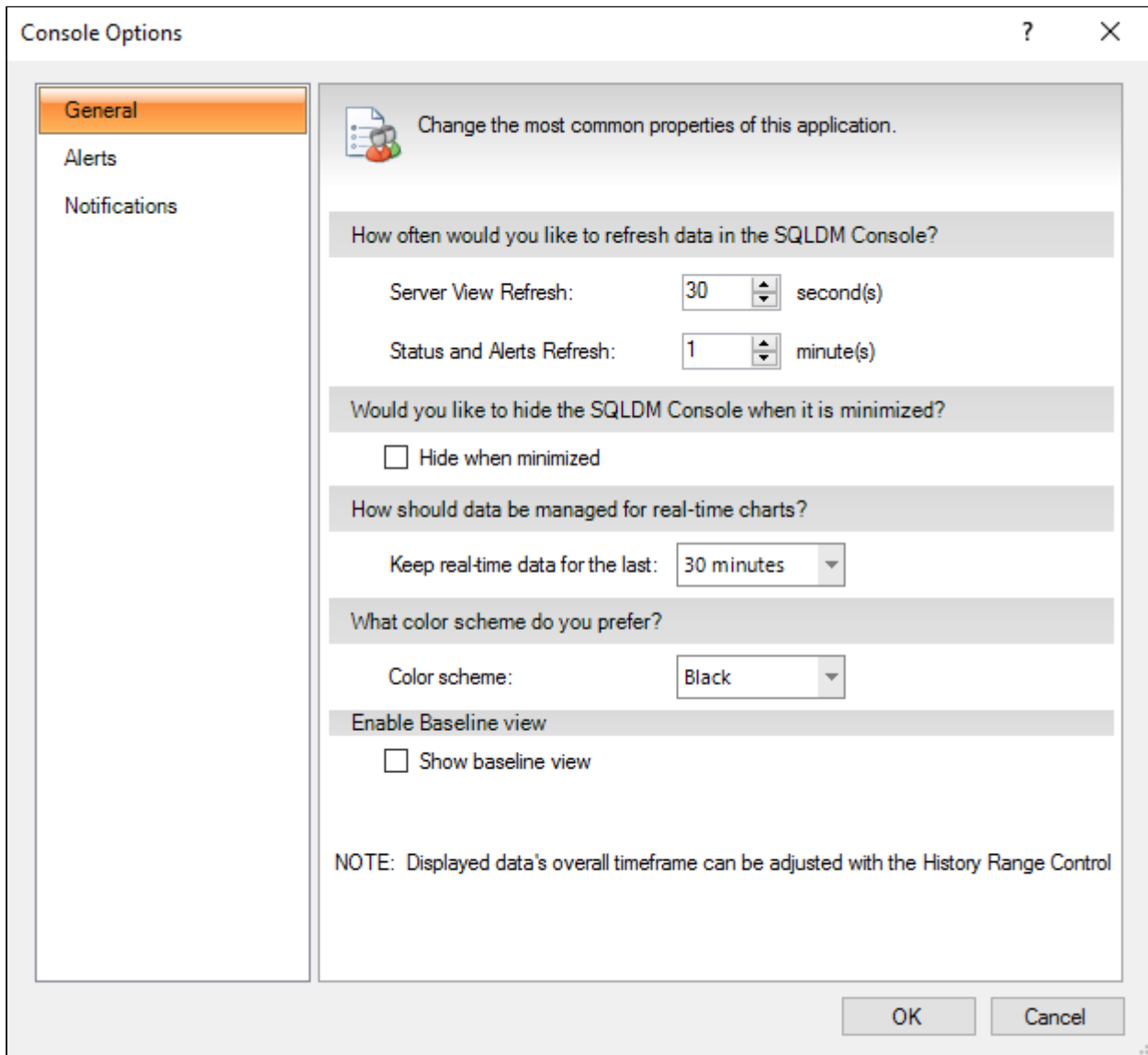
The [Alerts tab](#) allows you to limit the number of historical alerts displayed in the Alerts view, which can improve the performance of the view when there is a large number of alerts.

Notifications


The [Notifications tab](#) allows you to configure how SQL Diagnostic Manager notifies you when alert data crosses a threshold.

Define common properties

The options on the **General** tab represent the most common properties users can change in the SQL Diagnostic Manager Console.



Access the Console Options window General tab

You can open the General tab of the Console Options window from the SQL Diagnostic Manager Console by selecting  > **Console Options**. The General tab is available by default.

Set your refresh intervals

The Refresh Interval refers to the amount of time the SQL Diagnostic Manager Console waits before gathering data from the SQL Diagnostic Manager Repository. There are two types of refreshes. Each type is associated with different SQL Console components:

Server View Refresh

Refers to the amount of time the SQL Diagnostic Manager Console waits before refreshing the currently active view. When you first open a view, the view is refreshed with the latest data SQL Diagnostic Manager has collected directly from the monitored SQL Server instance. You may want to lower the amount of time between refreshes if you are trying to diagnose a specific problem that requires a shorter interval. The interval for this field is between one and 300 seconds.

Status and Alerts Refresh

Refers to the amount of time the SQL Diagnostic Manager Console waits before polling every monitored SQL Server instance and displaying notification pop-ups, updating the monitored SQL Server instance tree with status changes, and updating the SQLDM Today view with status changes. The interval for this field is between one and 60 minutes.

Hide the desktop client

The **Hide when minimized** toggle determines whether SQL Diagnostic Manager displays as an active application in your Windows taskbar. When you want to open your SQL Diagnostic Manager Console again, double-click the **IDERA SQL Diagnostic Manager** icon in your taskbar. Enabling or disabling this option has no effect on data collection or the refresh rate of your currently active view.

Use chart options

Chart options allow you to determine how much data is kept in your real-time charts and select from the following options: 1 minute, 5 minutes, 10 minutes, 15 minutes, 30 minutes, 1 hour, 2 hours, 4 hours, 8 hours, 12 hours, and one day.

Use a color scheme

The **Color scheme** field allows you to select the background color you prefer for the SQL Diagnostic Manager Management Console. By default, the scheme is set to **Black**.


Enable Baseline view

The Baseline view can be enabled and disabled by checking the "Show baseline view" box.

Configure the Alerts view

The Alerts tab of the Console Options window allows you to enter the number of rows you want displayed in the historical Alerts view. While this restriction limits the historical view, there is no limit to the number of rows used for your active alerts. The default entry is **0**, which does not limit the number of rows in your historical view.

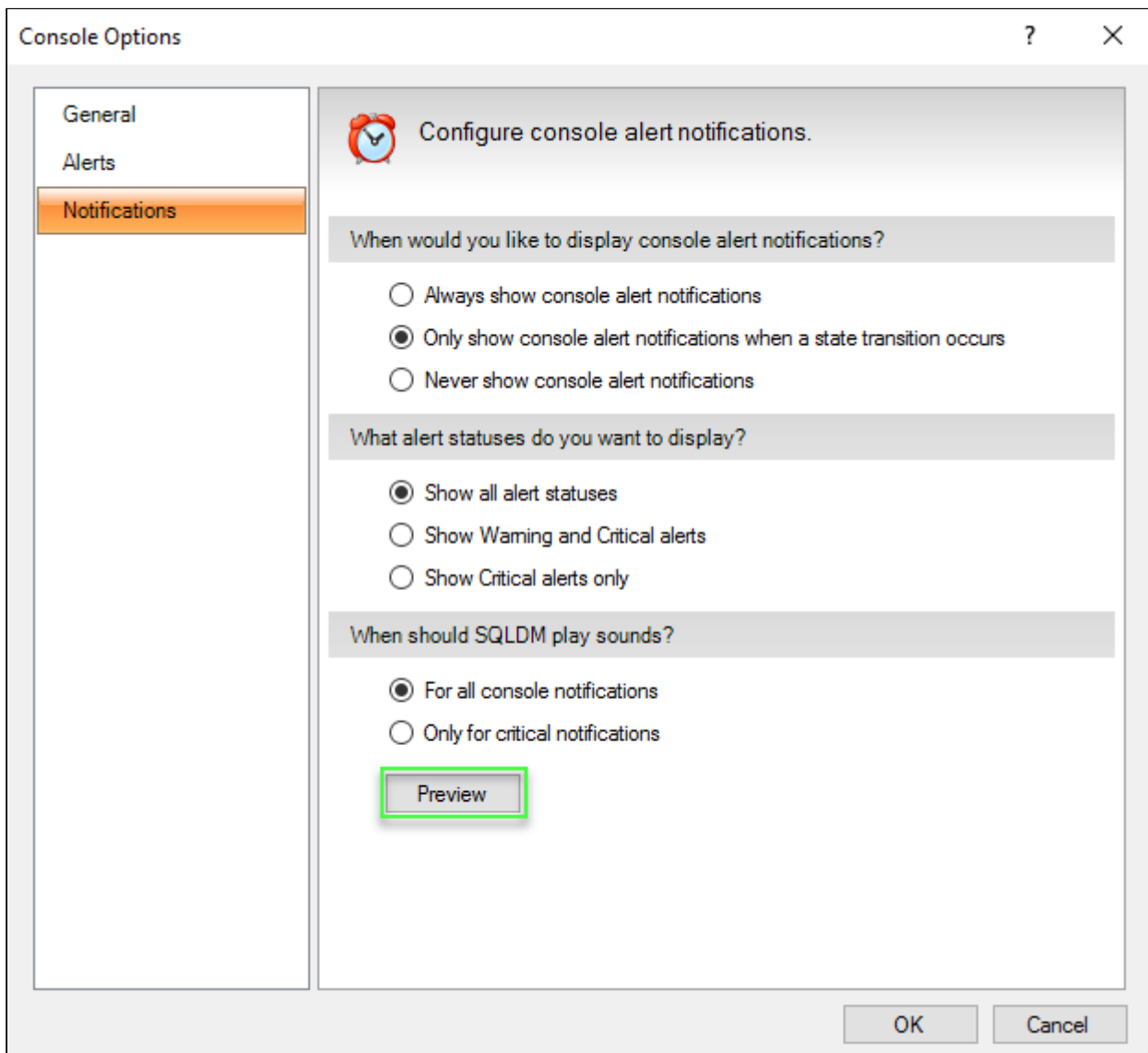
Access the Console Options window Alerts tab

You can open the Alerts tab of the Console Options window from the SQLDM Console by selecting  > **Console Options**, and then selecting the Alerts tab.


Configure notification settings

The Notifications tab of the Console Options window allows you to configure when and how the SQL Diagnostic Manager Console displays alert notifications. When you select an option to show notifications, SQL Diagnostic Manager displays a pop-up notification in your task bar each time it reaches a threshold.

To see an example of a console alert, click **Preview**, and the alert pop-up appears as shown in the following image.



Access the Console Options window Notifications tab

You can open the Notifications tab of the Console Options window from the SQL Diagnostic Manager Console by selecting  > **Console Options**, and then selecting the Notifications tab.

Selecting visual notifications

SQL Diagnostic Manager allows you to select when and for which alert states you receive alert notifications. You can select from the following options regarding when you want SQL Diagnostic Manager to display notifications:

Always show console alert notifications

Provides alert notifications occur after every refresh.

Only show console alert notifications when a state transition occurs

Limits the alert notifications you receive on refreshes to only those that change states.

Never show console alert notifications

Prevents the occurrence of all console notifications, but you continue to receive all email alerts. This setting is dependent upon your particular needs and you can modify the alert at any point. If you receive too many notifications, consider limiting notifications to occur only after an alert state transition. However, if you monitor a problematic SQL Server and want to know when a threshold is met, consider monitoring all alert states.

Select from the following options regarding which alert statuses trigger notifications:

Show all alert statuses

Display notifications for all alert state changes.

Show Warning and Critical alerts

Display notifications only when an event triggers a Warning or Critical alert.

Show Critical alerts only

Display notifications only when an event triggers a Critical alert.

Selecting audible notifications

The options in the When should SQL Diagnostic Manager play sounds section allow you to toggle whether you want to hear an audible notification along with the alert in SQL Diagnostic Manager whenever an event triggers a critical alert. The default **For all console notifications** option helps to let you know of any notification within SQL Diagnostic Manager. Select **Only for critical notifications** and SQL Diagnostic Manager plays the audible notification only when an event triggers a critical alert.

Set your SQL Diagnostic Manager Repository grooming schedule

The Grooming Options window allows you to schedule and configure SQL Diagnostic Manager Repository grooming. Grooming is the process of deleting metrics stored in the SQL Diagnostic Manager Repository that are older than a given date and time. It is important to groom data from your Repository to optimize the performance of processes such as report generation and data collection.

- ✔ You can limit the amount of data stored in the SQL Diagnostic Manager Repository by lowering the number of days data is stored. This is particularly important when [monitoring a large number of servers](#) or when storing Query information at a low threshold.

Access the Grooming Options window

To open the Grooming Options window, select  > **Grooming Options** from the toolbar menu.

In the Grooming Options window you can set an specific hour or determine after how many hours you want to groom your data and aggregate queries.

Also, you can define when your data should be groomed to save repository space, under the following options (in days):

- Groom standard metrics and baselines older than
- Groom Sessions, Queries, Deadlocks, Waits, and History Browser data older than
- Groom inactive Alerts older than
- Aggregate query data into daily records after
- Groom Change Log data older than
- Groom Prescriptive Analysis data older than
- Aggregate forecasting data to daily records after
- Groom forecasting data older than

Grooming Options ? X

i Depending on the amount of data collected, grooming can be a performance intensive operation. It is recommended that grooming be scheduled for a time when the repository is less active. This time is in the Repository's time zone.

The current time on the repository is 11/23/2018 10:48:34 AM.

Data Grooming	Aggregation
Delete old data and perform table and index maintenance.	Combine older data into daily roll-up records to save space.
<input checked="" type="radio"/> Once daily at 03:00 AM	<input checked="" type="radio"/> Once daily at 02:00 AM
<input type="radio"/> Every 4 hours	<input type="radio"/> Every 4 hours

When should data be groomed to save repository space?

i The amount of data collected and stored by SQLDM can be large depending on the number of SQL Server instances being monitored and how often data is collected. Use the following options to tailor data retention to your needs.

Groom standard metrics and baselines older than	1095	days
Groom Sessions, Queries, Deadlocks, Waits, and History Browser data older than	31	days
Groom inactive Alerts older than	31	days
Aggregate query data into daily records after	14	days
Groom Change Log data older than	90	days
Groom Prescriptive Analysis data older than	90	days
Aggregate forecasting data to daily records after	1095	days
Groom forecasting data older than	1095	days

What is the current grooming status?

Data Grooming	Aggregation
Currently running: No	Currently running: No
Last completed at: 11/23/2018 3:00:00 AM	Last completed at: 11/23/2018 2:00:01 AM
Completion status: Succeeded	Completion status: Succeeded

Refresh

Groom Now Aggregate Now **OK** Apply Cancel

The current grooming status is shown at the bottom of the Grooming Options window, it displays the current status of the grooming and the aggregation actions. It also displays the date and time of the latest operation, and the current completion status. Click **Refresh** to get the updated information of these actions at any time.

Click the options **Groom Now** and **Aggregate Now** located at the bottom of the Grooming Options window, to perform these actions instantly.

Use custom counters to track metrics

SQL Diagnostic Manager, displays a wide variety of the most common SQL Server and Operating System performance metrics, by default,. In addition to providing these common metrics, custom counters provide the ability to add additional performance metrics to the SQL Diagnostic Manager Console used for historical trending, on-demand monitoring, and schedule refresh alerting.

✓ Import or export a custom counter

In the [IDERA community site](#) you can [share](#) or [import](#) custom counters created by other SQL Diagnostic Manager users.

You can find a list of operating system counters by searching for "Windows Performance Counters" on technet.microsoft.com.

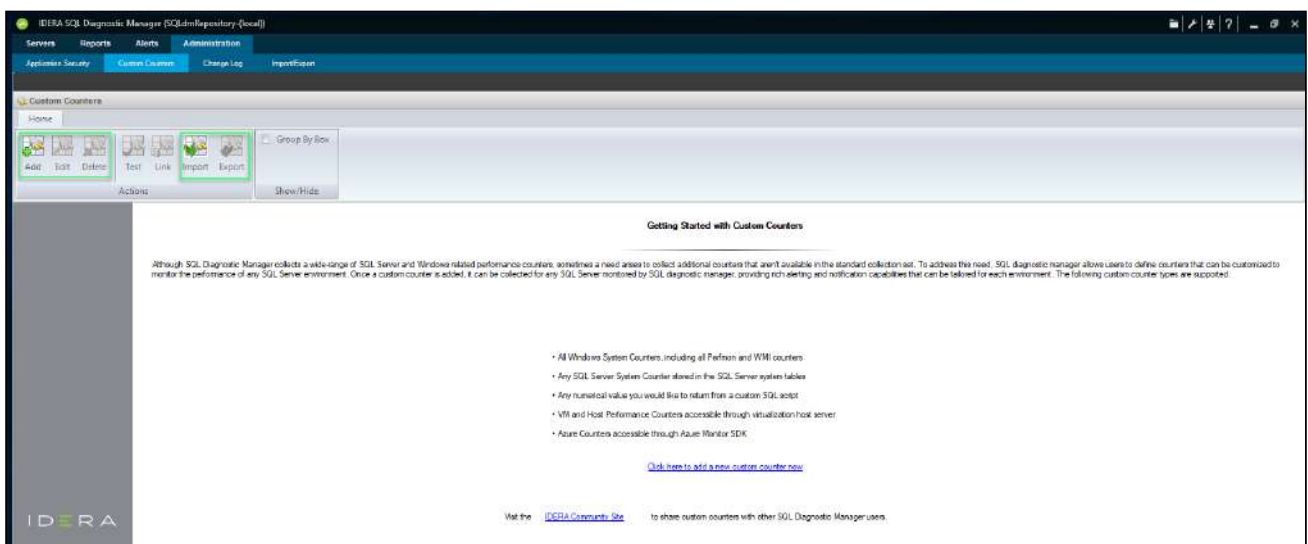
You can add the following types of counters:

- All Windows System Counters, including all Perfmon and WMI counters
- Any SQL Server System Counter stored in the sysperfmio system table
- Any numerical value you would like to return from a custom SQL script
- VM and host server performance counters accessible through the host Server
- Azure Server System Counter to monitor counters that are collected using API supplied by Azure.

✓ In addition to providing all of the above counters, you can add any additional counters contained in a new Windows or SQL Server Service Pack. This allows you the flexibility to monitor any metric that is important to your enterprise.

Access the Custom Counters view

To open the Custom Counters view, click **Administration > Custom Counters**.



Create a custom counter

To add a custom counter:

1. Select the **Administration** tab and click **Custom Counters**.
2. Click **Add**, to create a Custom Counter.
3. Click **Next** on the Add Custom Counter wizard Welcome page.
4. Select a counter type, and click **Next**.
5. Select the counter you want to monitor, and click **Next**.
6. Select the counter calculation type and scale factor, and click **Next**.
7. Type a name for the custom counter, select a **Category**, add a **Description**, and click **Next**.
8. Configure your alert settings related to the new custom counter, and click **Next**.
9. Click **Finish**.

Using custom SQL scripts

Use custom SQL scripts to monitor metrics that are not connected to system or SQL counters, under-utilized SQL 2005 DMVs, or metrics that are very specific to your environment, such as row count for a production table.

- ✔ SQL Diagnostic Manager does not scrub custom SQL batches for SQL Injection protection. Use caution when entering custom SQL scripts as it is an advanced feature. The go command is not a Transact-SQL statement and you cannot use this command in custom SQL batches. The go command is a batch separator for Management Studio, Query Analyzer, sqlcmb, and osql.

Locate your custom counters

Custom counters appear in the following places in the SQL Diagnostic Manager Console:

Alert Configuration window

Your new custom counters appear in the Alert Configuration and the Default Alert Configuration windows in the Custom category. You can edit the alert thresholds, and even decide whether you want an alert associated with the new counter.

Server Details view

The Server Details view lists all the metrics monitored by SQL Diagnostic Manager, including all your new custom counters. You can filter this view to display only your custom counters by selecting the Custom Counters radio-button in the Filter ribbon. You can access this window by selecting details on the Server view or by double-clicking Custom Counters in the Servers tree.

Custom Counters table

SQL Diagnostic Manager provides a list of all your custom counters in the Custom Counters table on the Administration window. In addition to viewing a list of custom counters, you can add a new counter, edit a counter, delete a counter, test a counter, or link a counter to a SQL Server instance for monitoring.

Add a custom counter

Custom counters are Windows Operating System or SQL Server metrics that are not monitored, by default, by SQL Diagnostic Manager. In addition to Windows Operating System and SQL Server metrics, you can enter T-SQL scripts to monitor any variable in your environment or select from a list of virtualization performance counters if you have monitored servers hosted on a virtual machine. Once added to the SQL Diagnostic Manager interface, you can monitor the results, view historical trending of the metric, configure alerts and receive notifications on them, and generate reports based on collected data.

Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters**, and selecting **Add** in the Custom Counters view.

Select counter type

The Select Counter Type window of the Add Custom Counter wizard allows you to choose the method you want SQL Diagnostic Manager to use when collecting the counter. SQL Diagnostic Manager offers the following counter types:

Windows System Counter

Select this option to pick from counters that include both those available in [Windows System Performance Monitor \(PerfMon\)](#) and [Windows Management Instrumentation \(WMI\)](#) performance counter types.

SQL Server System Counter

Select this option to pick from counters that are gathered via standard T-SQL from SQL Server.

Custom SQL Script

Select this option to enter a custom T-SQL script to monitor variables such as those which are not connected to the operating system or SQL Server counters or metrics that are very specific to your environment such as a row count on a particular production table. Note that your script must return a single numeric value.

Virtual Machine Counter

Select this option to pick from counters collected using an API from VMware or Hyper-V.

Azure Server System Counter

Select this option to pick from counters collected using API supplied by Azure.

Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters**, and clicking **Add** in the Custom Counters view.

Select Windows system counter

The Select Windows System Counter window of the Add Custom Counter wizard allows you to select the counter you want from a list populated from those currently available on an existing monitored SQL Server instance, or you can enter the counter information manually.

Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters**, and clicking **Add** in the Custom Counters view.

Select a counter from the list

As you select from a populated list from those currently available on an existing monitored SQL Server instance, the next drop-down list populates (filtered by your selection). This allows you to quickly find the specific counter you want to add to SQL Diagnostic Manager.

1. Select whether you want to populate the list with counters that are contained in PerfMon, or populate it with every WMI counter.
2. Pick the SQL Server instance to populate counter information from the drop-down list.
3. Select the object name from the drop-down list.
4. Select the name of the specific counter you want to add from the drop-down list.
5. Select the instance from the drop-down list. The instance name is used to distinguish objects of a given type. For example, `_Total` refers to all instances, a drive letter such as `C:` for a specific disk drive, or the name of an application.

✔ Click the **Details** button for information on the object name and counter name you have selected.

Manually add a counter

If you have the counter information available, you can enter the information quickly by simply pasting or typing the Object Name, Counter Name, and Instance name into their associated fields. The counter information entered must be the exact complicated name or the counter does not work correctly.

For example, to enter the Processor object name (as seen in PerfMon), use the following programmatic name entry depending on the system configuration:

```
Win32_PerfFormattedData_PerfOS_Processor
```

✔ You can find custom counter programmatic names on [Microsoft TechNet](#).

Enable SQL Diagnostic Manager Performance counters

Follow these steps to enable SQL Diagnostic Manager Performance Counters:

1. Go to the SQLDM installation folder path: C:\Program Files\Idera\Idera SQLdiagnostic manager
2. Open SQLdmManagementService.exe or SQLdmCollectionService.exe as Administrator (depending on which SQLDM Performance Counters you want to enable)
3. Edit it and add the following parameter: *publishPerformanceCounters="true"*, restart the services.

Review the following examples:

SQLdmCollectionService.exe

```
<configuration>
<Idera.SQLdm>
<Services>

<CollectionService
instanceName="Default" managementServiceAddress="ServerName"
publishPerformanceCounters="true"/>


</Services>
</Idera.SQLdm>
</configuration>
```


SQLdmManagementService.exe

```
<configuration>
<Idera.SQLdm>
<Services>

<ManagementService instanceName="Default"
repositoryServer="SQLInstanceName\Name"
servicePort="5166" repositoryDatabase="SQLdmRepositoryName"
publishPerformanceCounters="true"/>

</Services>
</Idera.SQLdm>
</configuration>
```

 If you no longer want to publish the SQLDM Performance Counters, set the parameter as follows *publishPerformanceCounters="false"* or delete the property from the configuration and restart the services.

 Review the available internal SQLDM PerfMon counters list for the SQLdmManagementService and SQLdmCollectionService: [Internal SQLDM perfmon counters](#).

Add custom counter alerts

Follow these steps to add internal SQLDM performance custom counter alerts:

1. Go to the Administration tab in the SQL Diagnostic Manager desktop and select Custom Counters.
2. Click **Add** to start configuring the custom counter.
3. In the Welcome screen of the Custom counter Wizard click **Next**.
4. Select the Windows System Counter type, and click Next.
5. Under the Select Windows System Counter screen, select the **Select a counter from an existing monitored SQL Server host (Recommended)** and in the Select Counter check the **Show only those counters available in Windows System Performance Monitor (Perfmon) (Recommended)** option.
 - a. In the SQL Server drop down list, select the SQL Server instance where you would like to add the custom counter.
 - b. In the Object Name field, select the `SQLdmCollectionService` or `SQLdmManagementService` option to retrieve the internal counters.
 - c. Choose the counter you preferred from the Counter Name drop down list and click **Next**. Review the list of [Internal SQLDM perfmon counters](#).
 - d. Select an instance or the instances if applied.
 - e. Click **Next**.
6. Under the Customize Counter Value screen, choose the Calculation Type by selecting either the **Use collected value** or the **Use per second value since last collection** (By default, the value for a custom counter is interpreted as the value collected). Also, Customize the Scale Factor by selecting the value from the drop down list.
 - a. After configuring the values Click **Test...** to open the Test Custom counter window.
 - b. In the Test custom Counter window, select the instances to test the custom counter that is being configured, and click **Test**. You can review the result of the test under the Result column.
 - c. Click **Done**.
7. Under the Customize Counter Definition step, type a name for the counter. By default, these counters are under the Custom Category but you can select an existing category or enter a category of your own. Click **Next**.
8. Under the Configure Alert Settings screen, select how your counter value should be evaluated for alerting: **Higher values are worse than lower values** or **Lower values are worse than higher values** .
 - a. Configure your Alert Thresholds, the values you provide for Informational, Warning, and Critical Thresholds are used as defaults and can be customized for each monitored instance.



When you click **Advanced**, you can generate choose to generate an alert or not when the value for the custom counter fails to be collected.

b. Check the box to enable the counter alert by default when is linked to a monitored SQL Server. Click **Next**.

9. In the Completing the Custom Counter Wizard screen, click **Finish** to complete the configuration of your custom counter.

When you complete the Custom Counter configuration, the "Would you like to link your counter to monitored SQL Server instances now?" message pops up, select **Yes** or **No** to continue:

- If you click Yes, the Link Custom Counter window opens. This window includes the Counter Summary information and the Linked Servers, you can select a SQLDM tag or specific monitored SQL Server instances, when you complete the configuration of the linked servers click **OK**.

Internal SQLDM perfmon counters

SQLdmManagementService Counters

- Active Workers
- Avg. Task Time
- Blocking Statistics inserts/sec
- Blocking Statistics sec/insert
- Custom Statistics inserts/sec
- Custom Statistics sec/insert
- Database Statistics inserts/sec
- Database Stats sec/insert
- Deadlock inserts/sec
- Deadlock sec/insert
- Disk Statistics inserts/sec
- Disk Statistics sec/insert
- ESX Config Data Statistics inserts/sec
- ESX Config Data Statistics sec/insert
- ESX Statistics inserts/sec
- ESX Statistics sec/insert
- Lock Statistics inserts/sec
- Lock Statistics sec/insert
- Mirror Participants grooming ops/sec
- Mirror Participants grooming sec/op
- Mirror Preferred Config ops/sec
- Mirror Preferred Config sec/op
- Mirror Replication Topology ops/sec
- Mirror Replication Topology sec/op
- Mirroring Statistics inserts/sec
- Mirroring Statistics sec/insert
- OS Statistics inserts/sec
- OS Statistics sec/insert
- Query Monitor Statistics inserts/sec
- Query Monitor Statistics sec/insert
- Query Wait Statistics inserts/sec
- Query Wait Statistics sec/insert
- Replication topology distributor inserts/sec
- Replication topology distributor sec/insert
- Replication topology inserts/sec
- Replication topology sec/insert
- Replication topology subscriber inserts/sec
- Replication topology subscriber sec/insert
- Server Statistics inserts/sec
- Server Stats sec/insert
- Session And Locks Statistics inserts/sec
- Session And Locks Statistics sec/insert
- Task Queue Length
- Tasks Queued/sec
- Temp Database File inserts/sec
- Temp Database File sec/insert
- VM Config Data Statistics inserts/sec
- VM Config Data Statistics sec/insert
- VM Statistics inserts/sec
- VM Statistics sec/insert
- Wait Statistics Details inserts/sec

- Wait Statistics Details written/sec
- Wait Statistics inserts/sec
- Wait Statistics written/sec
- Wait Type Statistics inserts/sec
- Wait Type Statistics written/sec
- Waiting Workers

SQLdmCollectionService\$Default

- #Monitores Servers
- #Servers in Maintenance Mode
- Active Workers
- Avg. Task Time
- Task Queue Length
- Tasks Queued/sec
- WaitingWorkers

SQLdmCollectionService\$Default\$PERINSTANCE

- #Alerts Raised
- #Databases
- #Dropped Scheduled Refreshes
- #Failed Collections
- #Query Monitor Statements
- #Queued Scheduled Refreshes
- #Wait Statistic Statements
- Collection Run Time

Select SQL Server system counters

The Select SQL Server System Counter window of the Add Custom Counter wizard allows you to manually enter counter information or select counters from a list populated from those currently available on an existing monitored SQL Server.

If you have the counter information available, you can enter the information quickly by simply pasting or typing the **Object Name**, **Counter Name**, and **Instance** name into their associated fields. The counter information entered must be exact or the counter does not work correctly.

To select a SQL Server system counter:

1. Select the appropriate SQL Server instance from the drop-down list.
2. Select the object name.
3. Select the name of the specific counter you want to add.
4. Select the instance, if applicable. The instance name is what distinguishes it from other similar objects.

If no counters appear in the drop-down list, either the counters are not installed or they are disabled, possibly by a recent Windows Service Pack.

Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters**, and selecting **Add** in the Custom Counters view.

How to calculate this metric

For more information about the algorithm used to calculate a specific metric, see [SQL Diagnostic Manager cannot load the performance counters needed for the statistics view pane - Solution #00000133](#) IDERA Solution available at the [Customer Support Portal](#).

Understand why the SQL Server counter grows

Many [SQL Server counter values](#) start at zero when a server starts and then grow persistently from there, including most of the values in the sysperfinfo system table.

For example, the Buffer Manager Page Lookups/Sec counter in the value collected mode always shows the total number of page lookups since the server was first started, while when in the value per second mode, it shows the number of lookups per second during the monitored period.

Provide custom SQL script

The Provide Custom SQL Script window of the Add Custom Counter wizard allows you to enter or paste a T-SQL script for your custom counter. For more information about T-SQL scripts, see [SQL Server Books Online](#).

Refer to the following tips when using custom SQL Scripts:

- Dangerous or destructive T-SQL that is entered as a custom counter is executed as entered, which could result in a negative impact on the linked monitored SQL Server instances.
- The `go` command is not a Transact-SQL statement and you cannot use this command in custom SQL batches. The `go` command is a batch separator for Management Studio, Query Analyzer, `sqlcmd`, and `osql`.
- Custom counter scripts run at every scheduled refresh and on-demand on the Custom Counters view. For best performance, custom scripts should run in under 30 seconds. If the script takes longer than two minutes to execute, timeouts may occur.
- Custom SQL scripts must return a single result set with a single numeric field. Additional fields or non-numeric values result in an error.
- Custom SQL scripts are displayed in Queries tab with "User Defined Counter" in their application name.
- The following options are set by default for custom counter scripts. If you need to use a different value, particularly another transaction isolation level, you should add the appropriate SET statement to the beginning of the script.
 - `set transaction isolation level read uncommitted`
 - `set lock_timeout 20000`
 - `set implicit_transactions off`
 - `set language us_english`
 - `set cursor_close_on_commit off`
 - `set query_governor_cost_limit 0`
 - `set numeric_roundabort off`

Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters**, and then clicking **Add** in the Custom Counters view.

Select virtual machine system counter

The Select Virtual Machine System Counter window of the Add Custom Counter wizard allows you to select the counter you want from a list populated by those counters currently available on an existing monitored SQL Server host server, or you can enter the counter information manually.

✔ You can select virtual machine counters for both the Hyper-V and VMware server virtualization platforms.

Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters** and selecting **Add** in the Custom Counters view.

Select the counter from the list

As you select from a populated list from those currently available on an existing monitored SQL Server host, the next drop-down list populates (filtered by your selection). This allows you to quickly find the specific counter you want to add to SQL Diagnostic Manager.

Add Custom Counter

Select Windows System Counter
Select the Windows System Counter you would like to monitor

Select a counter from an existing monitored SQL Server host **(Recommended)**
 Manually enter counter configuration

Select Counter

Show only those counters available in Windows System Performance Monitor (Perfmon) **(Recommended)**
 Show all Windows Management Instrumentation (WMI) counters

SQL Server: DESKTOP-9EFPMF4

Object Name: PhysicalDisk

Counter Name: < Select a counter >

Instance Name: < Select a counter >

- % Disk Read Time
- % Disk Time
- % Disk Write Time
- % Idle Time
- Avg. Disk Bytes/Read
- Avg. Disk Bytes/Transfer
- Avg. Disk Bytes/Write
- Avg. Disk Queue Length
- Avg. Disk Read Queue Length
- Avg. Disk sec/Read
- Avg. Disk sec/Transfer
- Avg. Disk sec/Write
- Avg. Disk Write Queue Length
- Current Disk Queue Length
- Disk Bytes/sec
- Disk Read Bytes/sec
- Disk Reads/sec
- Disk Transfers/sec
- Disk Write Bytes/sec
- Disk Writes/sec
- Split IO/Sec

1. Pick the monitored SQL Server instance to populate counter information from the drop-down list.
2. Select the object name from the drop-down list.
3. Select the name of the specific counter you want to add from the drop-down list.
4. Select the instance (if applicable) from the drop-down list. The instance name is what distinguishes it from other similar objects.

Add a counter manually

If you have the counter information available, you can enter the information quickly by selecting **Manually enter counter configuration**, and simply pasting or typing the name of the object, counter, and SQL Server instance into the associated fields. The counter information entered must be exact or the counter does not work correctly.

Specify counter value

The Custom Counter Value window of the Add Custom Counter wizard allows you to select the type of calculation used by your counter and set a scale factor so that your counter information displays values in their correct format. While the **Scale** drop-down list displays the same scale factors available in PerfMon, you can also type in a custom scale factor to use.

Custom Calculation Type

Choose to either have your counter display the value collected or calculate the value per second between refreshes.

Customize Scale Factor

Select a custom scale to convert the value into a more manageable number.

Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters** and clicking **Add** in the Custom Counters view.

Scale factor examples


The following table covers some conversion factors that allows you to precisely convert common units:

Collected Value	Desired Value	Scale Factor
Kilobyte	Byte	1024
Megabyte	Byte	1048576
Megabyte	Kilobyte	1024
Gigabyte	Kilobyte	1048576
Terabyte	Megabyte	1048576
Terabyte	Gigabyte	1024
Byte	Kilobyte	0.0009765625
Byte	Megabyte	0.000000953674316
Kilobyte	Megabyte	0.0009765625
Kilobyte	Gigabyte	0.0009765625
Megabyte	Gigabyte	0.0009765625

Collected Value	Desired Value	Scale Factor
Megabyte	Terabyte	0.000000953674316
Gigabyte	Terabyte	0.0009765625
Kilobyte	SQL Server Page	0.125
Megabyte	SQL Server Page	128
Gigabyte	SQL Server Page	131072
SQL Server Page	Kilobyte	8
SQL Server page	Megabyte	0.007813
SQL Server Page	Gigabyte	0.000007629395
Kilobyte	SQL Server Extent	0.015625
Megabyte	SQL Server Extent	16
Gigabyte	SQL Server Extent	16384
SQL Server Extent	Kilobyte	64
SQL Server Extent	Megabyte	0.0625
SQL Server Extent	Gigabyte	0.000061035156

Specify counter definition

The Customize Counter Definition window of the Add Custom Counter wizard is where you name, assign to a category, and enter a description of your new custom counter. The information entered on the Custom Counter Definition window appears in various places in the SQL Diagnostic Manager interface.

 You cannot change the name or category of a counter once you create it.

Name

The name you give your new custom counter is the most important way to distinguish your custom counters. You should give them a name that directly relates to the information monitored.

Category

The category you assign to your new custom counter is important for organizing your custom counters, if you have many, on the Custom Counters window and on the Alert Configuration window.

Description

The description you give your custom counter is an important way of distinguishing your custom counters if the name you give them does not provide enough of a difference.

Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters** and clicking **Add** in the Custom Counters view.

Configure alert settings for this counter

The Configure Alert Settings window of the Add Custom Counter wizard allows you to configure how SQL Diagnostic Manager uses and monitors your new custom counter information.

Configure Alert Evaluation

This option allows you configure whether lower values are worse than higher values such as the percent of free disk space on a drive, or higher values are worse than lower values such as the percentage of free memory used.

Configure Alert Thresholds

The informational, warning, and critical thresholds you enter dictate the values that trigger alerts. Once configured, you can change these values on the [Alert Configuration window](#).


Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters** and clicking **Add** in the Custom Counters view.

Configure a custom response when SQL Diagnostic Manager generates an alert

SQL Diagnostic Manager allows you to configure when and how alerts generate responses. You can select from the following alert responses (providers):

- Send an email to a person or group of people
- Log an event in the Windows Event Log
- Enable the Query Monitor
- Execute a program or utility
- Send an event to your Network Management tool
- Execute a SQL Agent Job
- Execute a Script

To configure a custom response when an alert is generated, select  **> Alert Actions and Responses** from the console toolbar.

Select Azure Server System Counter

The Azure Server System counter allows SQL Diagnostic Manager to monitor counters that are collected using API, which is supplied by Azure.

After you select the Azure Server System Counter type when adding a Custom Counter, you need to provide some configuration details of your Azure server:


1. Select a server from the drop-down list of the Azure Server option.
2. Select a profile from the drop-down list of the Azure Profile option.
3. Select a type from the drop-down list of the Resource Type option.
4. And, select a name from the drop-down list of the Resource Name option.

To view the Azure profiles that are registered in SQL Diagnostic Manager click **View Profiles**.

To complete the addition of the Azure Server System Counter, select the metric details from the drop-down metric list.

 Azure Monitor Custom Counter Limitations

- The Azure Counter import is disabled, by default.
- The linking of the Azure Custom Counter is not supported.
- The Azure Counters are only visible in monitored Azure servers.
- The Azure Counters are not supported for tags.

 The 'Monitor ManagementClient.Metrics' library collects data for the Azure Custom Counters. It takes input parameters like metric name and interval and returns values for the metrics in that particular interval. However, based on the Azure instance activity at that interval, the returned value of the metric is null when those values are zero or inexistent.

Azure Profiles

The Azure Profiles Configuration window allows you to view the registered Azure profiles and the details of how these profiles are linked. Also, it allows you to add, view/edit, remove, and test the accessibility of the Azure subscriptions.

1. Under the Application Profiles section, click **New**.
2. In the Azure Application Profile, select a Name, and add a Description.
3. Under the Azure subscription section, click **New** to open the Azure Subscription configuration window, enter the Subscription, add a description, and click **OK**.
4. Under the Azure Application section, click **New**, fill in the fields in the Azure Application Configuration window: Name, Tenant Id, Client Id, Secret, and Description. Click **OK**.


To review or edit any information of the Azure Linked Profiles, click **View/Edit** to open the Azure Linked Profile window which allows you to edit or view the Azure Profile description, select the Azure Server, and select the Application Profile.

Additionally, you can access the available resources to link by clicking **Available Resources**.

1. In the Available Azure Resources window, select an Azure server from the drop-down list.
2. Select an Application Profile from the drop-down list to review all the available resources for the selection.

Edit a custom counter

The Edit Custom Counter wizard allows you to edit your custom counter settings. To edit a custom counter, select the counter from the list on the Custom Counters window and click the **Edit** button.

 You cannot change the custom counter name and category once you create a Counter.

Delete a custom counter

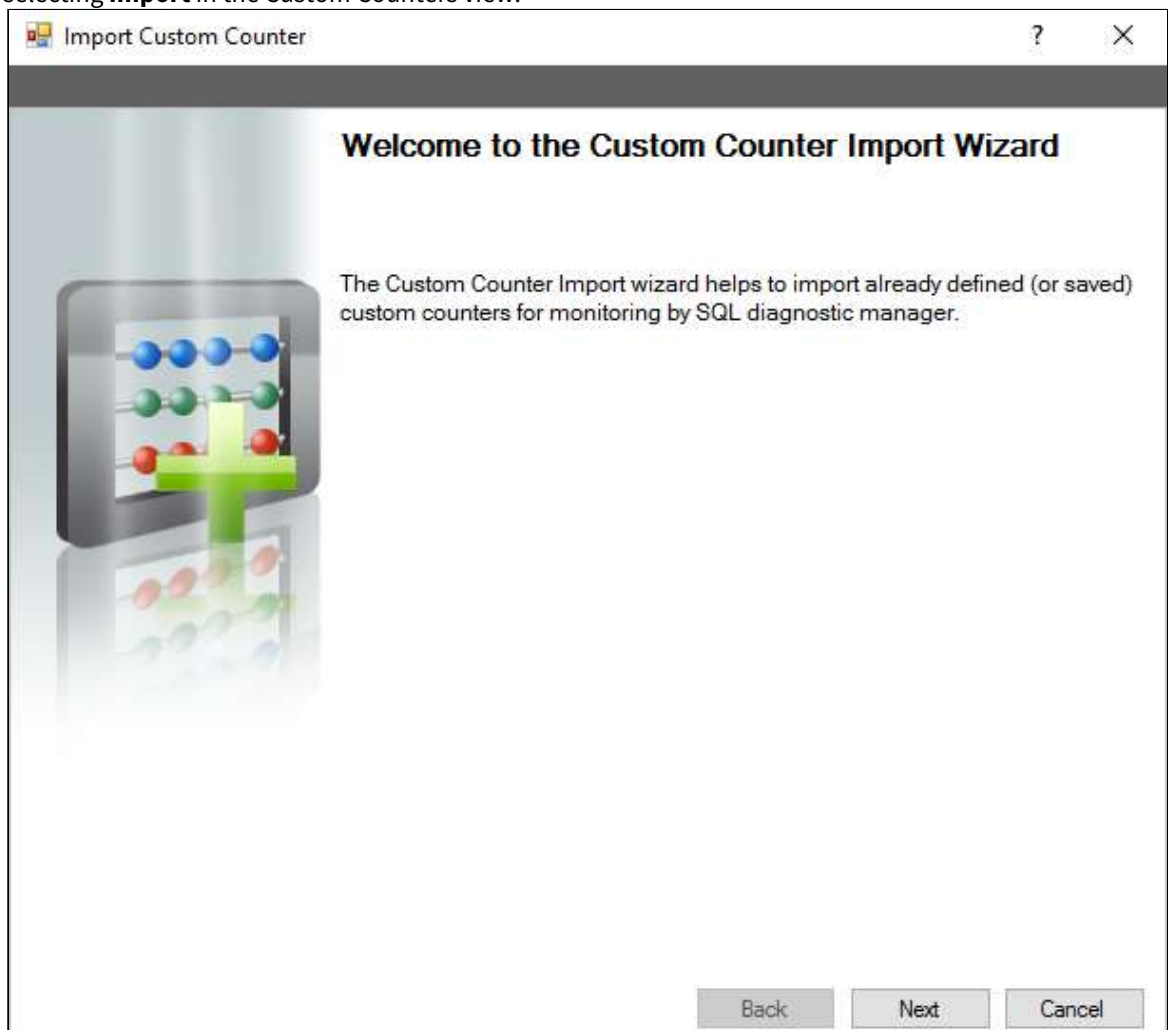
Deleting a custom counter from SQL Diagnostic Manager removes it from the interface and stops collection. Select a custom counter on the **Custom Counters** window and click **Delete** to remove it from SQL Diagnostic Manager.

Import a custom counter

Custom counters are Windows Operating System or SQL Server metrics that are not monitored, by default, by SQL Diagnostic Manager. You can import already created custom counters from the [IDERA community site](#) through the Custom Counter Import Wizard. Once added to the SQL Diagnostic Manager interface, you can monitor the results, view historical trending of the metric, configure alerts and receive notifications on them, and generate reports based on collected data.

To import a custom counter:

1. Open the Custom Counter Import Wizard by clicking **Administration > Custom Counters**, and selecting **Import** in the Custom Counters view.



2. On the **Welcome to the Custom Counter Import** page, click **Next**.

✔ You can also access the Custom Counter import wizard by clicking **Administration > Import/Export** from the navigation pane and selecting **Custom Counter** in the import column.

3. On the **Select custom counters to import** page, browse for custom counters' XML files to import.
4. On the **Summary of imported custom counters** page, you can verify that the selected counters were imported correctly or if there are any errors you can view details by clicking **View validation errors** or review any steps by clicking **Back**.

5. Click **Finish** to import the valid custom counters and exit the wizard.



Export a custom counter

For instructions on exporting a custom counter, see [Export a custom counter](#).

Export a custom counter

Custom counters are Windows Operating System or SQL Server metrics that are not monitored, by default, by SQL Diagnostic Manager. You can export and share already created custom counters in the [IDERA community site](#) through the **Actions** group on the **Home** tab or through the Import/Export wizard.

To export a custom counter:

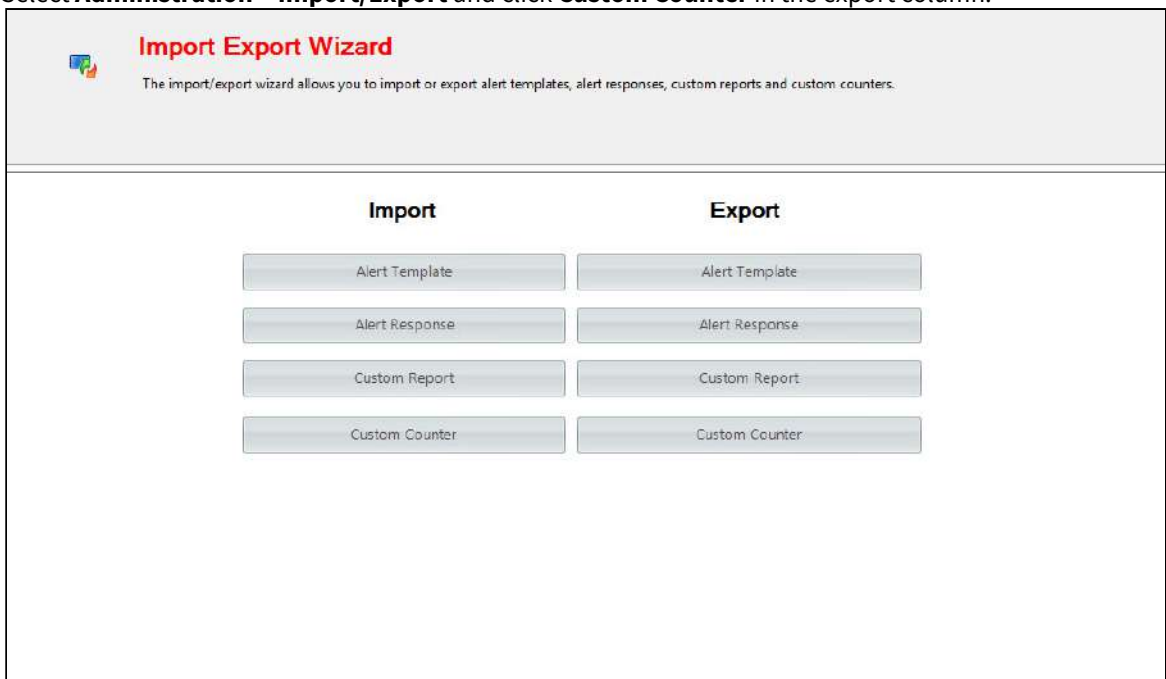
1. Select **Administration** in the navigation pane.
2. Select **Custom Counters**.
3. Select the custom counter you wish to export from the list and click **Export** in the **Actions** group.
4. In the **Browse For Folder** dialog box, indicate in which folder you want to save the custom counter XML file.
5. In the "Selected custom counters exported to the selected directory" notification, click **OK**.

✔ Import a custom counter

For instructions on importing a custom counter, see [Import a custom counter](#).

To export all your custom counters:

1. Select **Administration > Import/Export** and click **Custom Counter** in the export column.



2. In the **Browse For Folder** dialog box, indicate in which folder you want to save the custom counters XML files.
3. Click **OK** in the "Custom counters exported to selected directory" notification.

Test a custom counter

The Test Custom Counter window allows you to verify that the custom counter is what you want to collect on the SQL Server instances you have linked it to and that the value returned is the value expected. A successful test displays the value of the metric collected, while a failed test provides the associated failure details.

- ✔ Common causes for failed custom counter tests include inaccurate counter information entered manually and counters that do not exist on a particular SQL Server instance.

Link custom counter to monitored instance

The Link Custom Counter window allows you to select the monitored SQL Server instances upon which you want your custom counter to appear. Select [Server tags from the Tags drop-down list](#) or [choose New tag](#) from the list.

Use the **Add** and **Remove** buttons to select the SQL Server instances to link to your custom counter and click **OK** to save your changes.

Automatically add a custom counter to a new registered SQL Server instance

Yes. By using links and tags, you can set SQL Diagnostic Manager to automatically add any custom counter to a newly-registered SQL Server instance.

To set SQL Diagnostic Manager to automatically add a customer counter to a new registered SQL Server instance:

1. [Create a custom counter](#) or [import a custom counter](#) if the appropriate counter does not exist.
2. [Create a tag](#) if the appropriate tag does not exist.
3. Link a custom counter to a tag by selecting the custom counter, and then clicking **Link**. You can include more than one custom counter per tag.
4. Select the tag you want to use, and then click **OK**. The custom counter is now associated with the tag.
5. Select the tag you just linked when adding a new SQL Server instance.

Work with tags


Tags are customizable labels that allow you to organize and view your monitored resources in a manner best suited to your needs. You can assign tags to monitored SQL Server instances, custom counters, and permissions.

By creating and assigning tags to different objects, you can better manage how SQL Diagnostic Manager presents information. For example, you may create an "Accounting" tag for all SQL Server instances used by the Accounting department. You could then copy alert configuration and assign custom counters and permissions to all the instances labeled "Accounting."

Tags are also accessible to all users. This gives you greater freedom, allowing users to work with the same tags across all SQL Diagnostic Manager consoles.

Manage tags


The Manage tags window allows you to [Add](#), [Edit](#), and Remove Tags. For more information on tags and their functionality, see [Working with tags](#).

 This window is available only to SQL Diagnostic Manager Administrators.

The Manage Tags window shows a list of all existing tags and you can perform the following actions:

- Modify the list of servers, custom counters, and permissions associated with a tag.
- [Create a new tag](#).
- [Edit an existing tag](#).
- Remove a tag (Removing a tag does not delete items associated with that tag).

Access the Manage Tags window

To access the Manage Tags window, select  > **Manage Tags**.

Add a new tag

The **Add Tag** window allows you to create and assign a tag while adding an instance. Tags are labels that allow you to organize server instances into meaningful groups.

Tags are customizable for your needs. For example, you may want to view only server instances used by a particular department, in a specific location, or by some other method, such as SQL Server version. You can assign multiple tags to each instance so that the instance appears in several different groups.

You may select one or more existing tags from the list, or create new ones by clicking **Add Tag**.

Edit a tag

The Tag Properties window allows you to view and change the properties of an existing Tag. To access the Tag Properties window, right-click a Tag and select **Tag Properties**.

The following information is available and editable in the Tag Properties window:

Servers

The Servers tab lists the SQL Server instances monitored by SQL Diagnostic Manager and allows you to select or deselect the servers included in the tag.

Custom Counters

The Custom Counters tab allows you to choose the custom counters you want to include in your tag.

Application Security

The Application security tab allows you to select the application security permissions you want to apply to your tag.


User Creation Scripts

SQL Diagnostic Manager includes scripts, which are used to create new users with the required permissions for the product to function. For more information, review the following links:

- [User Creation Scripts On-Premises](#)
- [User Creation Scripts Azure SQL](#)


User Creation Scripts Azure SQL

The minimum permission required to run a user creation script for Azure SQL is Server Admin rights.

 For more information about the characteristics of the Server admin and Azure Active Directory admin accounts, review [Controlling and granting database access to SQL Database and SQL Data Warehouse](#).

The user creation scripts for Azure SQL are detailed, as follows:

- Minimum permissions Azure SQL
- Recommended permissions Azure SQL


 The Azure SQL recommended permissions for managed instances in SQLDM are the same as [On-Premises recommended permissions](#).

Minimum Permissions Azure SQL

SQL Diagnostic Manager now includes a SQL script to create a new user with minimum permissions for the product to function:

1. Use the Find/Replace function to locate and replace all references of 'SQLdmConsoleUser' with the name of your SQLdm Monitor User.
2. Connect to the Azure SQL Server, execute the first section of the script against the master database and the second section of the script against the target databases.

Find the **MinimumPermissionsAzure.sql** script below:


File	Modified
 MinimumPermissionsAzure.sql	Nov 17, 2020 by IDERA


Recommended Permissions Azure SQL

SQL Diagnostic Manager now includes a SQL script to create a new user with minimum permissions for the product to function:

1. Use the Find/Replace function to locate and replace all references to 'SQLdmConsoleUser' with the name of your SQLdm Monitor User.
2. Connect to the Azure SQL Server and execute the first section of the script against the master database and the second section of the script against the target databases.

Find the **RecommendedPermissionsAzure.sql** script below:

File	Modified
 RecommendedPermissionsAzure.sql	Nov 17, 2020 by IDERA

 Review the [Minimum and Recommended Permissions for Azure SQL](#) for more details on the permissions you need to create a new user for Azure SQL Database.

User Creation Scripts On Premises

SQL Diagnostic Manager allows you to monitor SQL Server instances without SysAdmin rights. However, these lowered permissions limit the availability of data.


The user creation scripts are detailed, as follows:

- [Recommended permissions on-premises](#)
- [Minimum permissions on-premises](#)

Recommended permissions on-premises

SQL Diagnostic Manager also includes a SQL script to create a new user with minimum permissions necessary for the product to function.

1. Use the Find/Replace function to locate and replace all references of 'SQLdmCollectionUser' with the SQL user that is to be used for monitoring the SQL Server instance.
2. Connect to the SQL Server that you want to monitor with the Admin user and run this script to create the SQL user with the recommended permissions to view the data on most of the screens of the application. Find the **RecommendedPermissionsOnPremises.sql** script below:

File	Modified
 RecommendedPermissionsOnPremises.sql	Aug 16, 2018 by IDERA

3. When registering the instance in SQL Diagnostic Manager, specify the SQL account that was created through the script so that SQL Diagnostic Manager connects to the monitored instance using the newly created account.

Recommended Permissions

These are the recommended permissions on-premises:

GRANT SERVER LEVEL PERMISSIONS

```
/** Assign VIEW SERVER STATE Permissions to SQLdmCollectionUser **/  
GRANT VIEW SERVER STATE TO [SQLdmCollectionUser]
```

```
/** Assign VIEW ANY DEFINITION Permissions to SQLdmCollectionUser **/  
GRANT VIEW ANY DEFINITION TO [SQLdmCollectionUser]
```

```
/** Assign VIEW ANY DATABASE Permissions to SQLdmCollectionUser **/  
GRANT VIEW ANY DATABASE TO [SQLdmCollectionUser]
```

GRANT ADDITIONAL XEVENTS AND TRACE PERMISSIONS

```
/** Assign ALTER TRACE Permissions to SQLdmCollectionUser **/  
GRANT ALTER TRACE TO [SQLdmCollectionUser]
```


```
/** Assign ALTER ANY EVENT SESSION Permissions to SQLdmCollectionUser **/  
GRANT ALTER ANY EVENT SESSION TO [SQLdmCollectionUser]
```

```
GRANT CREATE ANY DATABASE TO [SQLdmCollectionUser]
```

Minimum permissions on-premises

SQL Diagnostic Manager now includes a SQL script to create a new user with minimum permissions for the product to function.

1. Use the Find/Replace function to locate and replace all references of 'SQLdmCollectionUser' with the SQL user that is to be used for monitoring the SQL Server instance.
2. Connect to the SQL Server that you want to monitor with the Admin user and run this script to create the SQL user with the minimum permissions required to view the data on most of the screens of the application. Find the **MinimumPermissionsOnPremises.sql** script below:

File	Modified
 MinimumPermissionsOnPremises.sql	Aug 16, 2018 by IDERA

3. When registering the instance in SQL Diagnostic Manager, specify the SQL account that was created through the script so that SQL Diagnostic Manager connects to the monitored instance using the newly created account.

Welcome to SQL Diagnostic Manager

SQL Diagnostic Manager is a powerful performance monitoring and diagnostics solution that proactively alerts administrators to health, performance and availability problems within the SQL Server environment. SQL Diagnostic Manager now integrates with the IDERA Dashboard, a common technology framework, designed to support the IDERA product suite. Users are able to obtain an overview of the status of their SQL Servers and hosted databases all in a consolidated view and navigate to individual product dashboards for details. The IDERA Dashboard provides a central set of services for managing users, product registry, instance registry, aggregated alerts across IDERA applications, a central web server, and tags for grouping instances. For additional information, see [Navigate the IDERA Dashboard](#).

To get started, see the following topics for more information:

- [Upgrade to this build of SQL Diagnostic Manager](#)
- [Configure SQL Diagnostic Manager](#)
- [Migrate your SQL Diagnostic Manager installation](#)

What is SQL Diagnostic Manager?

SQL Diagnostic Manager provides an unprecedented level of diagnostic information on the health, performance, and status of SQL Server instances across your environment. You can view, diagnose, and report on critical performance statistics from a central point of control. Using a unique agentless architecture, SQL Diagnostic Manager gathers diagnostic information in real time, keeping database administrators and managers informed by providing customized alerts. Easy to install and use, SQL Diagnostic Manager is an indispensable SQL Server management tool that benefits both experienced and junior database administrators.

What is the IDERA Dashboard?

The IDERA Dashboard is a common technology framework designed to support the entire IDERA product suite. The IDERA Dashboard allows users to get an overview of the status of their SQL Servers and hosted databases all in a consolidated view, while providing users the means to drill into individual product dashboards for details. The IDERA Dashboard supports multiple copies of SQL Diagnostic Manager installations.

The SQL Diagnostic Manager web console is integrated with the [IDERA Dashboard](#). It is designed to provide quick access to the overall status of your SQL Server environment to help diagnose any issues. The web console provides key information simply through dashboards and high-level views to a large user base such as IT, helpdesk, and DBAs. You can access the web console from any connected browser and grant access as you need. To learn more about the SQL Diagnostic Manager web console, visit [Navigate the web console dashboard](#).

What is SQL Diagnostic Manager Mobile?

SQL Diagnostic Manager Mobile is a Web application that displays real-time SQL Server performance dashboards on a variety of mobile devices, including iPhone, iPad, Android, and Blackberry. SQL Diagnostic Manager Mobile includes the IDERA Newsfeed technology. The IDERA Newsfeed is a revolutionary new way for DBAs and managers to collaborate, share knowledge, and keep close tabs on their most critical SQL Server issues.

How does SQL Diagnostic Manager help me?

Whether you need immediate diagnostic information or want to gather data for long-term capacity planning, you can use SQL Diagnostic Manager to help:

- Increase SQL Server availability
- Provide a comprehensive view of current performance
- Provide continuous, unattended monitoring
- Store data in the centralized SQLdm Repository database
- Improve the productivity of both skilled and novice database administrators

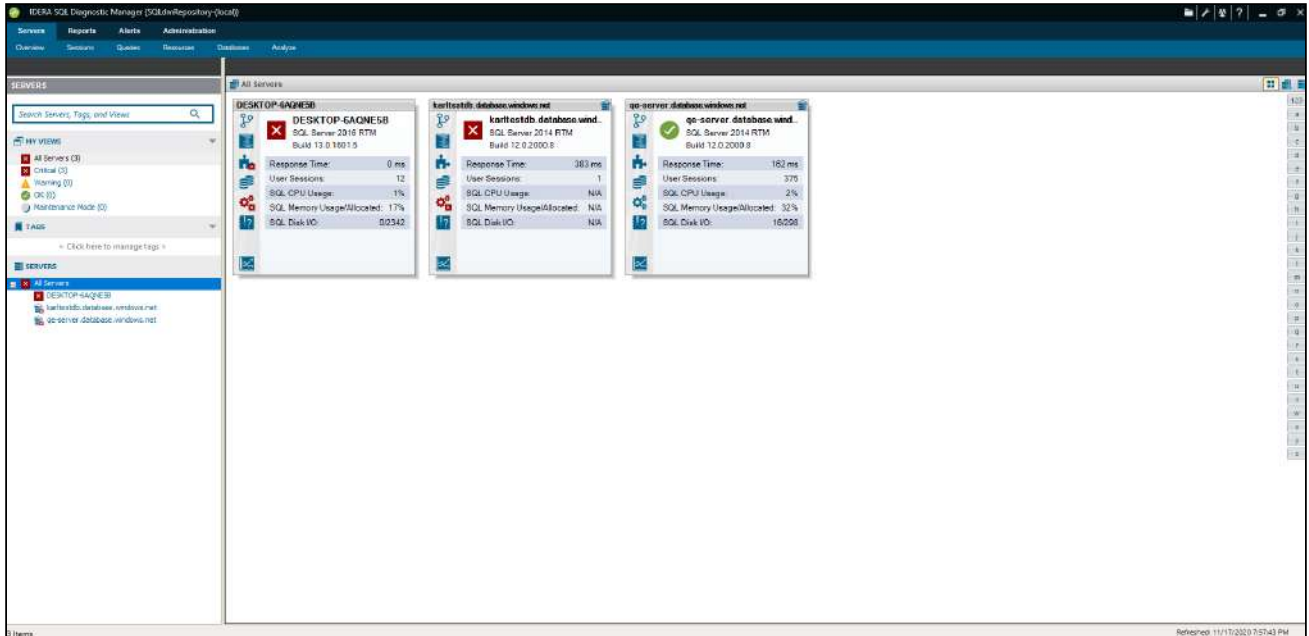
With clear, easy-to-understand charts and reports, you can track standard SQL Server performance metrics, trend historical performance data over time, and proactively identify potential performance problems.

Use the SQL Diagnostic Manager interface

SQL Diagnostic Manager uses a graphically enhanced user interface to display the current state of your SQL Server instances. A mixture of graphs, charts, and color coding makes problem identification quick and easy, allowing you to resolve issues before they reach critical levels.

Navigation pane

The Navigation pane displays on the left side of the SQLDM Console as shown in the following image:



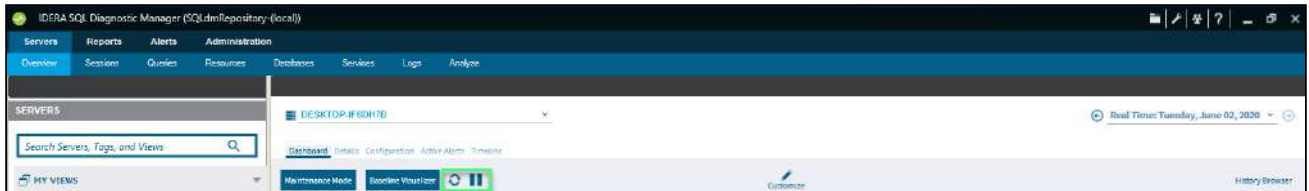
Servers

The MY VIEWS section helps you organize your SQL Server instances by status and includes any custom views you have set up. The Tags section allows you to manage tags or key words that you can use to quickly find the appropriate monitored SQL Server instances, custom counters, and permissions. The Servers section lists the SQL Server instances contained in the currently selected view.

To view all registered instances, select **All Servers** from the MY VIEWS list, review the page [Monitor multiple SQL Server instances](#).

Refresh and pause views

To let you control the information refresh on a particular view, SQLDM includes the **Refresh** and **Pause** buttons as shown in the following image:



You can [edit the scheduled refresh of your views](#) for more current information. SQL Diagnostic Manager offers the following refresh controls:

Refresh button

The **Refresh** button tells SQL Diagnostic Manager to immediately collect data and refresh the current Console view. Depending on the view, the refresh could take several seconds to complete. SQL Diagnostic Manager displays the current refresh status in the lower-right corner of the SQLDM Console.

Pause button

The **Pause** button tells SQL Diagnostic Manager to stop refreshing the console. If you click **Pause** during a refresh, the pause occurs after the refresh completes. To continue refreshing, click **Pause** again. Pausing the refresh allows you to maintain the currently-monitored state so you can see what is happening on your SQL Server instance without incoming information overwriting the data. This feature also gives you time to export or print the displayed statistics before retrieving the next data set.

Charts

Charts are populated from the point at which you navigate to the window with the chart. This allows you to view real-time charts for the data point selected. Real-time viewing is dependent on your [refresh interval](#). Once populated, you can manipulate the type of chart you want to display and export the chart or copy it to the clipboard to paste into other programs.


- ✓ In the Server Details view, click the **Maximize** button on the upper-right corner of a chart to display a large version of the chart. The **Restore** button returns the chart to the default view.

All-time charts in the web console and desktop console have time-range drill-down functionality modeled after SQL Workload Analysis.

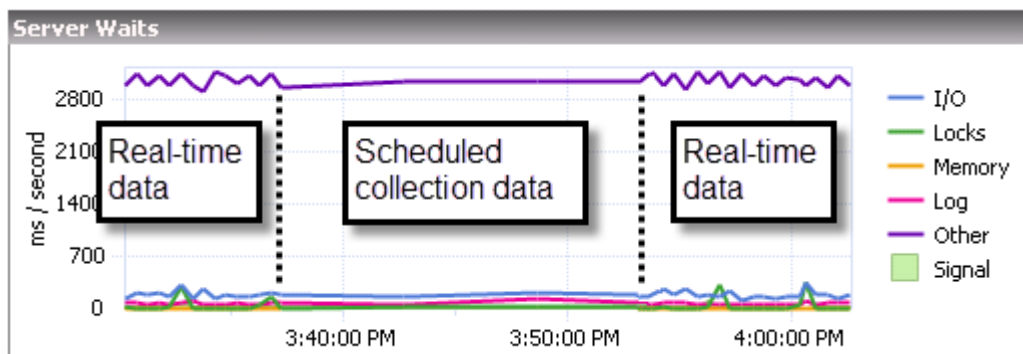
1. Click on your preferred graph and hold the left mouse button.
2. Drag to the left or to the right.
3. Release the mouse button.

The time range selected with the previous steps is the new display history range on all the graphs displayed on different tabs.

Understanding your chart data and refresh interval

SQL Diagnostic Manager charts provide data whether collected in real-time or as part of a scheduled collection event. When you open a view, SQL Diagnostic Manager begins populating the chart with real-time data collected on the interval specified in the **Server View Refresh** field on the [Console Options window](#) ( > **Console Options**).

If you pause or leave this view and return during the same Console session, SQL Diagnostic Manager displays the real-time data points as previously viewed, followed by a period of data points from any scheduled collections that occurred while you were away. The chart returns to populating with new real-time data points, as shown in the following image:



The displayed data is determined by the last range selected in the top right corner drop-down.

Customize your charts

SQL Diagnostic Manager contains charts that are populated by selecting the appropriate data items collected. Once populated, you can manipulate the type of chart you want to display and export the chart or copy it to the clipboard to paste into other programs.

You can personalize charts by right-clicking anywhere inside the chart and selecting **Toolbar** from the menu. The chart toolbar contains the following options:

Copy To Clipboard

The **Copy To Clipboard** button allows you to copy the chart to your Windows clipboard in various formats.

Print

The **Print** button allows you to print the chart.

Gallery

The **Gallery** menu includes a number of options that allow you to change the type of chart used to represent your data.

Anti-Aliasing

The **Anti-Aliasing** button smooths the lines between your data points to more easily view your data.

Palette Selector

The **Palette Selector** menu includes a number of color choices for you to change the color scheme of your chart.

3D/2D

The **3D/2D** toggle lets you choose to view your results in three- or two-dimensional plots. Additional options become active based on your choice and allow you to rotate your view or cluster your results on the z-axis.

Axes Settings

The **Axes Settings** menu lets you choose how to view each axis, whether to display labels and additional properties for your view.

Point Labels

The **Point Labels** toggle lets you choose to view the actual data at each point.

Data Grid

The **Data Grid** toggle places a table below your chart that displays the data included in your chart.

Legend Box

The **Legend Box** toggle lets you choose to view a legend for your displayed data.

Zoom

The **Zoom** button allows you to zoom in on a point in your chart.

Properties

The **Properties** button displays additional properties for further customization of your chart.

Monitor multiple SQL Server instances

MY VIEWS allows you to group servers to provide a high-level, global view of multiple monitored servers. This view includes basic statistics about your registered SQL Server instances and quick links to more detailed information. A monitor displays statistics for each SQL Server instance included in the group.

Access My Views

MY VIEWS is available at all times as part of the navigation pane as shown in the following image:

The screenshot displays the IDERA SQL Diagnostic Manager (SQLdmRepository-(local)) interface. The top navigation bar includes 'Servers', 'Reports', 'Alerts', and 'Administration'. Below this, a secondary bar shows 'Overview', 'Sessions', 'Queries', 'Resources', 'Databases', 'Services', 'Logs', and 'Analyze'. The main content area is split into two panes. The left pane, titled 'SERVERS', contains a search bar and a 'MY VIEWS' section with a list of groups: 'All Servers (3)', 'Critical (3)', 'Warning (0)', 'OK (0)', and 'Maintenance Mode (0)'. Below this is a 'TAGS' section and a list of servers including 'DESKTOP-IF6DH7B', 'SIDROCTST02\SQL2014', and 'SIDROCTST03\SQL2014'. The right pane shows a detailed view for 'DESKTOP-IF6DH7B' with tabs for 'Dashboard', 'Details', 'Configuration', 'Active Alerts', and 'Timeline'. It features a 'Maintenance Mode' button, a 'Baseline Visualizer' button, and a 'CPU' usage graph comparing 'SQL Server' (blue) and 'OS' (green) usage over time. A 'Memory' section below shows 'SQL Used', 'SQL Allocated', and 'Total Used'.

Use the group list

SQL Diagnostic Manager updates the data shown in the thumbnails view of the group list according to the interval entered in the **Status and Alerts Refresh** field on the [Console Options window](#), accessed by selecting > **Console Options**. This information is collected from the last data point stored in the SQLDM Repository.

By default, the group list displays basic statistics for each of the SQL Server instances monitored by SQL Diagnostic Manager. This information includes the following statistics for each instance:

Response Time

Refers to the time it takes for a query to go from the SQLDM Console to the SQL Server instance and back.

User Sessions

Refers to the total number of SQL Server user sessions.

SQL CPU Usage

Refers to average percentage of SQL Server processor usage on the computer hosting the SQL Server instance.

SQL Memory Usage/Allocated

Refers to amount of memory in use by the monitored SQL Server instance.

SQL Disk I/O

Refers to the number of physical reads and physical writes made by the SQL Server instance between refreshes. **If you see a high percentage of physical reads**, you can improve performance by allocating more memory to the SQL data cache.

Customize the monitors in your group list

SQL Diagnostic Manager allows you to customize the monitors displayed in group lists so that you display only those key statistics and charts that are most important to you.

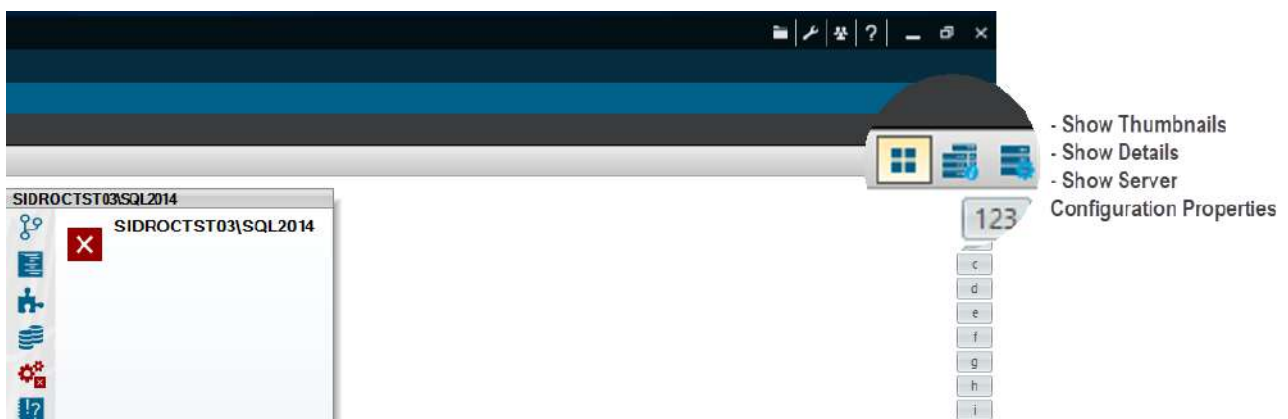
To customize a monitor:

1. Right-click a SQL Server instance in the All Servers screen.
2. Select **Thumbnail** and select the information type you want to display on your monitor.

Instead of listing statistics for the server, SQL Diagnostic Manager displays the individual chart of the data item you selected. To revert back, select **Summary** from the **Thumbnail** list.

Use the Server Thumbnails, Details, or Configuration Properties window

SQL Diagnostic Manager allows you to monitor multiple instances using different views, such as thumbnails of mini-monitors, a list of instances and their important details, or a list of instances and their server properties. SQL Diagnostic Manager defaults to the thumbnails view. To access a different view, click the appropriate button in the upper-right corner of the window as shown in the following image:




- ✔ If any of the metrics are missing from this view, you may need to [enable OS metrics collection](#).

These views list all the data collected at the last refresh for all your monitored SQL Server instances. The Details and [Server Configuration Properties](#) views can assist you with comparing the status of a specific metric or property between two or more of your SQL Server instances.

Add a custom view

You may want to add a custom view to the My Views list to organize your SQL Server instances into specific groups for monitoring purposes. This is especially important if you are monitoring a large number of SQL Server instances and want information for a specific group of SQL Server instances. If you create views before adding your SQL Server instances to SQL Diagnostic Manager for monitoring, you can have these new instances added to your views by selecting the new view you create and then adding the SQL Server instances.

To add a new view:

1. To create a new view from the Files menu, select  > **Create View**. If you want to create a new view from My Views, right-click within the My Views section, and select **Create View**.
2. Type a name for the view.
3. Select the monitored SQL Server instances you want to include in this view.
4. Click **Add >** to move the selected instances from **Monitored Instances** to **View Instances**.
5. Click **Create** to create your new view.

Server Configuration Properties

The Server Configuration Properties view provides the current configuration settings for each of your monitored SQL Server instances and allows for some changes directly in this window rather than leaving this view to access the properties page. The grid layout allows you to quickly scroll down in the column for a specific setting, and verify that the setting is correct in all of your instances.

You can sort your results in ascending or descending order for each column by clicking the column header. Right-click a column header to select other column options, including grouping by the selected column, hiding a column, or opening the **Group By** box for additional options.

Access the Server Configuration Properties view

SQL Diagnostic Manager provides two paths to access the Server Configuration Properties view. The first access path is by clicking **All Servers**, and then clicking the **Show Server Configuration Properties** icon on the far right. The second access path is by right-clicking **All Servers**, and then clicking **View All Server Properties**. By default, the All Servers view displays your monitored SQL Server instances in the thumbnail view.

Performing general tasks in this view

The Server Configuration Properties view provides the following actions by right-clicking a row, and then selecting the task you want to perform:

Open

Opens the Dashboard view for the selected monitored SQL Server instance.

Refresh Alerts

Gathers the alert status and refreshes the tree view information for the selected monitored SQL Server instance.

Delete

Allows you to delete the SQL Server instance from your SQLDM installation. When you select **Delete**, SQL Diagnostic Manager displays a message asking whether you want to retain the collected data for the SQL Server instance, and it also allows you to cancel out of the deletion process.



You cannot retrieve collected data once it is deleted from the SQLdm Repository. Use the **Delete** option with care.

Collapse All Groups

Allows you to collapse the contents of a group of monitored SQL Server instances.

Expand All Groups

Allows you to expand the contents of a group of monitored SQL Server instances.

Print

Allows you to print the grid of your monitored SQL Server instances and their configured properties.

Export To Excel


Allows you to export the grid of your monitored SQL Server instances and their configured properties to Microsoft Excel.

Properties

Opens the Monitored SQL Server Properties window for the selected instance.

Server Configuration Properties view contents

When you select the Server Configuration Properties view, SQL Diagnostic Manager displays a grid containing each of the settings for all of your monitored SQL Server instances. This information is updated each time the SQLdm Collection Service runs. Some cells allow you to make changes on this view while others are managed in other areas in SQL Diagnostic Manager, which are accessible by double-clicking the cell you want to change. SQL Diagnostic Manager displays the Change icon in the **Changed?** column for the row in which you made the change. Make sure you click **Save Changes** once you made the necessary change. Click **Reset Changes** to revert all unsaved changes you made on this view. The Change icon helps remind you to save your changes as shown in the following image.

All Servers								
Changed?	Server Name	SQL Version	SQL Edition	Authentication Mode	Check Server Accessibility	Alert Refresh	Database Statistics Refresh	Maintenance Mode
	SIDROCTST02\SQL2014	SQL Server 2014 RTM	Enterprise Edition (64-bit)	SQL Authentication: sa	00:00:50	00:04:00	01:00:00	Not set
	SIDROCTST03\SQL2014	SQL Server 2014 RTM	Enterprise Edition (64-bit)	SQL Authentication: sa	00:00:30	00:06:00	01:00:00	Not set

The Server Configuration Properties view contains the following properties:

Active Cluster Node

Displays the name of the active cluster node on this monitored SQL Server instance. For more information about this setting, see [Configure cluster settings](#).

Alert Refresh

Displays the interval (in hours, minutes, and seconds) between times when the service collects diagnostic data and raises the associated alerts. Use a value between 30 seconds and 30 minutes. Lower values result in a faster triggered alert, but also cause more frequent refreshes, which may increase monitoring overhead. You can modify this setting directly in this view. Click in the appropriate field, make your change, and then click **Save Changes**. For more information about this setting, see [Set general server options](#).

Authentication Mode

Displays the authentication mode used by the SQLdm Collection Service to collect diagnostic data from the monitored SQL Server instance. For more information about this setting, see [Set general server options](#).

Activity Monitor Enabled

Displays whether monitoring of non-query activities is enabled. The monitoring of non-query activities captures autogrow events, deadlocks, and blocks. For more information about this setting, see [Set activity monitor options](#).

Baseline Date Range

Displays the period of time over which the server baseline is calculated. For more information about this setting, see [Set server baseline options](#).

Baseline Time Period

Displays the date range over which the server baseline is calculated. For more information about this setting, see [Set server baseline options](#).

Changed?

Displays an edit icon to indicate that this row includes a change from the last time the data was saved. Click **Save Changes** to retain any changes made in this view. Click **Reset Changes** to revert to the most-recently saved server configuration properties.

Check Server Accessibility

Displays the interval (in hours, minutes, and seconds) between times when the server availability is verified. Use a value between 30 seconds and 10 minutes. If a select 1 query cannot execute in this timeframe, then the server is considered unresponsive. Setting your server availability check to a very low value may result in false positive alerts. You can modify this setting directly in this view. Click in the appropriate field, make your change, and then click **Save Changes**. For more information about this setting, see [Set general server options](#).

Database Statistics Refresh

Displays the interval (in hours, minutes, and seconds) between times when the database space-related data is collected and associated alerts are raised. Use a value between one minute and 24 hours. Setting your database statistics refresh to a low value may result in false positive alerts. A low setting also causes refreshes to occur more often, which increases the monitoring overhead. In environments with a large number of databases whose sizes do not change rapidly, setting this refresh to a long interval can greatly reduce your monitoring footprint. You can modify this setting directly in this view. Click in the appropriate field, make your change, and then click **Save Changes**. For more information about this setting, see [Set general server options](#).

Deadlock Monitoring

Displays the current state of deadlock monitoring, which raises alerts for deadlocked sessions on your monitored SQL Server instance. Deadlock monitoring is supported on monitored servers running SQL Server 2005 or greater and is dependent on enabling the monitoring of non-query activities in the Activity Monitor window. For more information about this setting, see [Set activity monitor options](#).

Disk Collection Settings

Displays whether the connected disk drives are discovered automatically. For more information about this setting, see [Customize disk statistics](#).

Error Log Alerting Enabled

Displays whether error log alerting is enabled, causing a specific error log collector to run. Error log alerting is considered enabled if either the SQL Server Agent Log or SQL Server Error Log alert is enabled. Note that these collectors may negatively affect performance. You can reduce the impact of error log reading by regularly cycling the server error logs. For more information about error log alerting, see [Configure alerts](#).

Fragmentation Min. Table Size (KB)

Displays the minimum size, in kilobytes, a table must meet before fragmentation statistics are collected. Use a value between 0 KB and 999999 KB. Fragmentation statistics are only gathered on tables with clustered indexes. You can modify this setting directly in this view. Click in the appropriate field, make

your change, and then click **Save Changes**. For more information about this setting, see [Edit table statistics options](#).

InputBuffer Limiter

Displays the limit of the executions performed by the DBCC Inputbuffer. This command retrieves the actual input command for the Session Details view, among others. Note that on busy servers, limiting this value can reduce monitoring impact. For more information about this setting, see [Set general server options](#).

Job Alerting Enabled

Displays whether job alerting is enabled, causing a specific job alerts collector to run. Job alerting is considered enabled if one of the following alerts is enabled:

- [SQL Server Agent Job Completion](#)
- [SQL Server Agent Job Failure](#)
- [SQL Server Agent Long Running Job \(Minutes\)](#)
- [SQL Server Agent Long Running Job \(Percent\)](#)

For more information about job alerting, see [Configure alerts](#).

Last Fragmentation Collection

Displays the date and time of the most recent, successful table fragmentation collection. This timestamp is local to the monitored SQL Server instance. For more information about this setting, see [Edit table statistics options](#).

Last Table Growth Collection

Displays the date and time of the most recent, successful table statistics collection. This timestamp is local to the monitored SQL Server instance. For more information about this setting, see [Edit table statistics options](#).

Linked Custom Counters (Count)

Displays the total number of custom counters monitored on this SQL Server instance. For more information about this setting, see [Link or tag custom counters](#).

Linked Custom Counters (List)

Displays the name for each custom counter monitored on this SQL Server instance. For more information about this setting, see [Link or tag custom counters](#).

Maintenance Mode

Displays the occasions when this server is in maintenance mode. Options include **Never**, **Until further notice**, **Recurring every week at the specified time**, and **Occurring once at the specified time**. For more information about this setting, see [Schedule maintenance mode](#).

OS Metrics Collection

Displays whether SQL Diagnostic Manager uses OLE automation or a direct connection to Windows Management Interface (WMI) to collect operating system and disk statistics. Disabling this setting prevents the collection of certain metrics in SQL Diagnostic Manager, but may lessen the monitoring overhead. For more information about this setting, see [Configure OS metrics](#).

Preferred Cluster Node

Displays the name of the preferred cluster node on this monitored SQL Server instance. You can modify this setting directly in this view. Click in the appropriate field, make your change, and then click **Save Changes**. For more information about this setting, see [Configure cluster settings](#).

Query Monitor Enabled

Displays whether query monitoring is enabled. The Query Monitor collects poorly-performing queries such as SQL batches, statements, stored procedures, and triggers. Low thresholds for query monitoring may impact performance. For more information about this setting, see [Set Query Monitor options](#).

Query Monitor Thresholds

Displays the various threshold types for query monitoring. Note that low thresholds for the query monitoring impacts performance. For more information about these settings, see [Set Query Monitor options](#).

Query Waits

Displays the date and time when the query-level wait statistics are collected or whether they are collected indefinitely. By default, these statistics are available only when the Query Waits view is open. Collection of query waits is a performance-intensive operation. For more information about this setting, see [Configure wait monitoring](#).

Query Waits Filters

Displays the filter configuration of the query-level wait statistics for this monitored SQL Server instance, including which applications, databases, and SQL text you want to exclude from the query wait statistics collection. Collection of query waits is a performance-intensive operation and excluding unnecessary data can reduce the monitoring footprint. For more information about this setting, see [Configure advanced query wait options](#).

Replication Monitoring Disabled

Displays whether replication statistics collection is disabled. Replication monitoring can cause excess monitoring overhead in some environments and is not necessary for all servers. You can modify this setting directly in this view. Click in the appropriate field to make your change, and then click **Save Changes**. For more information about this setting, see [Disable replication statistics collection](#).

Server Name

Displays the name of the monitored SQL Server instance.

Session History Browser Collection Disabled

Displays whether extended session data collection is disabled, including collection of session details, locks, and blocks. Collection of this data allows you to view past session detail, lock, and block information, but incurs extra monitoring overhead to provide this data. This data also requires extra space in the SQLdm Repository. You can modify this setting directly in this view. Click in the appropriate field to make your change, and then click **Save Changes**. For more information about this setting, see [Set general server options](#).

SQL Edition

Displays the edition of SQL Server software used in this instance.

SQL Server ID

Displays the identification number of the monitored SQL Server as used internally by SQL Diagnostic Manager. You cannot modify this value.

SQL Version

Displays the version of SQL Server software used in this instance.

Table Fragmentation Collection

Displays the days of the week and time of day when an attempt at table statistics collection occurs. For more information about this setting, see [Edit table statistics options](#).

Table Growth Collection

Displays the days of the week and time of day when an attempt at table statistics collection occurs. For more information about this setting, see [Edit table statistics options](#).

Table Statistics Excluded Databases (Count)

Displays the total number of databases excluded from table statistic collection. For more information about this setting, see [Edit table statistics options](#).

Table Statistics Excluded Databases (List)

Displays the name of each database excluded from table statistic collection. For more information about this setting, see [Edit table statistics options](#).

Virtualization Host Name

Displays the name of the host server on which this monitored SQL Server instance is running. This field applies only to SQL Server instances running on a virtual machine. For more information about virtualization configuration, see [Configure virtualization](#).

VM Name

Displays the name of the virtual machine on which this monitored SQL Server instance is running. This field applies only to SQL Server instances running on a virtual machine. For more information about virtualization configuration, see [Configure virtualization](#).

Use the SQL Diagnostic Manager web console

The SQL Diagnostic Manager web console helps users identify and diagnose problems quickly in their SQL Server environment. The SQL Diagnostic Manager web console provides key overview information simply through dashboards and high level views.

- Access the overall status of your SQL Server environment from any connected browser. **No SQL Diagnostic Manager Desktop Console required.**
- Provide an easy to use web console without giving access to the entire management console.
- Quickly view status of a single instance or instance groups (tags) and drill down for details.
- View all active alerts in your SQL Server environment and access detailed information.
- Select different sub-views to diagnose performance issues faster.
- Monitor overall health of queries and drill down through [advanced query views](#) for details.
- View your top values for selected metrics.

Find answers

This documentation set includes a comprehensive online Help system as well as additional resources that support you as you install and use the product. You can also search IDERA Solutions, available at the [IDERA Customer Service portal](#)

View help topics

SQL Diagnostic Manager Help supports the following web browser minimum requirements:

- Internet Explorer IE 9.x+
- Google Chrome
- Mozilla Firefox
- Edge Browser

Print help topics

You can print a help topic by selecting the topic in the table of contents, and then clicking the **Print this topic** icon on the right side of the Help window.

Use the PDF if you want to print more than a single topic. To print, click the PDF tab, and then click the **Print file** button. In the Print dialog box, select the pages you want to print before clicking **Print**.

Customer support portal

The IDERA Customer Support Portal offers 24x7 worldwide support to all customers who hold a current maintenance contract with IDERA and provides an easy and convenient way to:

- Submit or track technical support cases.
- Search the IDERA Knowledge Base.
- Download the latest version of your IDERA products.
- Access information about your support plan and licenses.
- Access IDERA resources such as the community site, blog, product videos, and white papers.

Access the IDERA Customer Support Portal

IDERA SQL product customers can click this link to enter the portal: <https://idera.secure.force.com/>

For any questions regarding access to the Customer Support Portal, contact your [IDERA Sales representative](#).

Support Status

For Support Status, please see the [SQL Tools Supported Versions](#) page.

Copy of Support Status

The table below provides a summary of the support status for SQL Diagnostic Manager. Please contact IDERA Customer Support for further information.

Version	Full Support	Limited Support	End of Life	Release	Limited Support	EOL Date
10.2	✓			June, 2017		
10.1	✓			October, 2016	12/31/2017	12/31/2018
10.0	✓			December, 2015	12/31/2017	12/31/2018
9.1	✓			May, 2015		
9.0	✓			February, 2015		
8.6	✓			October, 2014	12/31/2017	6/6/2018
8.5	✓			August, 2014	12/31/2017	6/6/2018
8.0	✓			October, 2013	12/31/2017	6/6/2018
7.5			✓	October, 2012	6/1/2016	12/30/2016
7.2			✓	March, 2012	6/1/2016	12/30/2016
7.1			✓	January, 2012	6/1/2016	12/30/2016
7.0			✓	October, 2011	6/1/2016	12/30/2016
6.6			✓	February, 2011	6/1/2016	12/30/2016
6.5			✓	November, 2010	6/1/2016	12/30/2016
6.2			✓	August, 2010		
6.1			✓	December, 2009		
6.0			✓	May, 2009		

Version	Full Support	Limited Support	End of Life	Release	Limited Support	EOL Date
5.7			✓	January, 2009		
5.6			✓	November, 2008		
5.5			✓	May, 2008		
5.0			✓	September, 2007		
Pre-5.0			✓			

Definitions

- Full Support.** Includes access to IDERA's SQLDM customer support website with online resources such as an extensive knowledge-base, product documentation, technical support via phone and online case tracking, software downloads, end of life, and product alerts. Service Packs, patches, and hotfixes are delivered for each fully supported version of software. Licensee may have to upgrade to the latest generally available release in order to receive the above resolution mechanisms. Please note, the forgoing is a general representation of IDERA's support terms and is to be used only for informational purposes.
- Limited Support.** Includes access to IDERA's SQLDM customer support website with online resources such as an extensive knowledge-base, product documentation, technical support via phone and online case tracking, software downloads, end of life, product alerts, and access to **existing** product Service Packs, patches and hotfixes. No new Service Packs, patches, or hotfixes will be provided for versions that are listed under Limited Support. Licensees who have paid all applicable Support and Maintenance fees are eligible for upgrading to the latest release. Please note, the forgoing is a general representation of IDERA's support terms and is to be used only for informational purposes.
- End of Life.** The process of retiring a product release. A product will reach its end of life either by subsequent newer release (obsolescence) or IDERA Product Management has decided to discontinue a product and no later version with the same functionality is commercially available from IDERA. Access to the latest release is provided to a Licensee with a current Support and Maintenance contract, but no support is provided for End of Life products.

Navigate the IDERA Dashboard

The IDERA Dashboard is a common technology framework designed to support the entire IDERA product suite. The IDERA Dashboard allows users to get an overview of the status of their SQL Servers and hosted databases all in a consolidated view, while providing users the means to drill into individual product dashboards for details. It also includes widgets that consolidate data across the products and supports multiple copies of SQL Diagnostic Manager installations. For more information, please review the [Manage the IDERA Dashboard](#).

The Administration view of the IDERA Dashboard provides a central set of services related to specific actions, register SQL Diagnostic Manager in the IDERA Dashboard review the [Register SQL Diagnostic Manager in the IDERA Dashboard](#).

Register SQL Diagnostic Manager in the IDERA Dashboard

The IDERA Dashboard hosts IDERA products that register with the Dashboard. Click the **Manage Products** option in the Administration view to open the respective window.

In the **Manage Products** window you can perform the following actions on registered products:

Edit a Product

To edit a product, select it from the Products list. The Dashboard displays the settings for the selected product on the panel to the right. Make the necessary modifications and click **Save**.

The screenshot shows the 'MANAGE PRODUCTS' interface. On the left, a table lists registered products:

Product	Version	Registered	Location	Credentials
IderaDashboard	4.4.0.118	Apr 23, 2018 11:29:39 AM	C:\B-2K&R2	Simpsons\Administrator
SQLdm(mn)	10.3.0.0	Apr 23, 2018 11:46:26 AM	local:\SQLdmRepository	Simpsons\Administrator

The right pane, titled 'Edit IDERA Registered Product', contains the following fields:

- Product: SQLdm(mn)
- Display Name: dm
- Version: 10.3.0.0
- Registered: Apr 23, 2018 11:46:26 AM
- Location: local:\SQLdmRepository
- Product Administrator Credentials:
 - User Name: simpsons\Administrator
 - Password: (To confirm enter the password here.)

Buttons at the bottom right include Cancel, Save, Delete, and Migrate product.

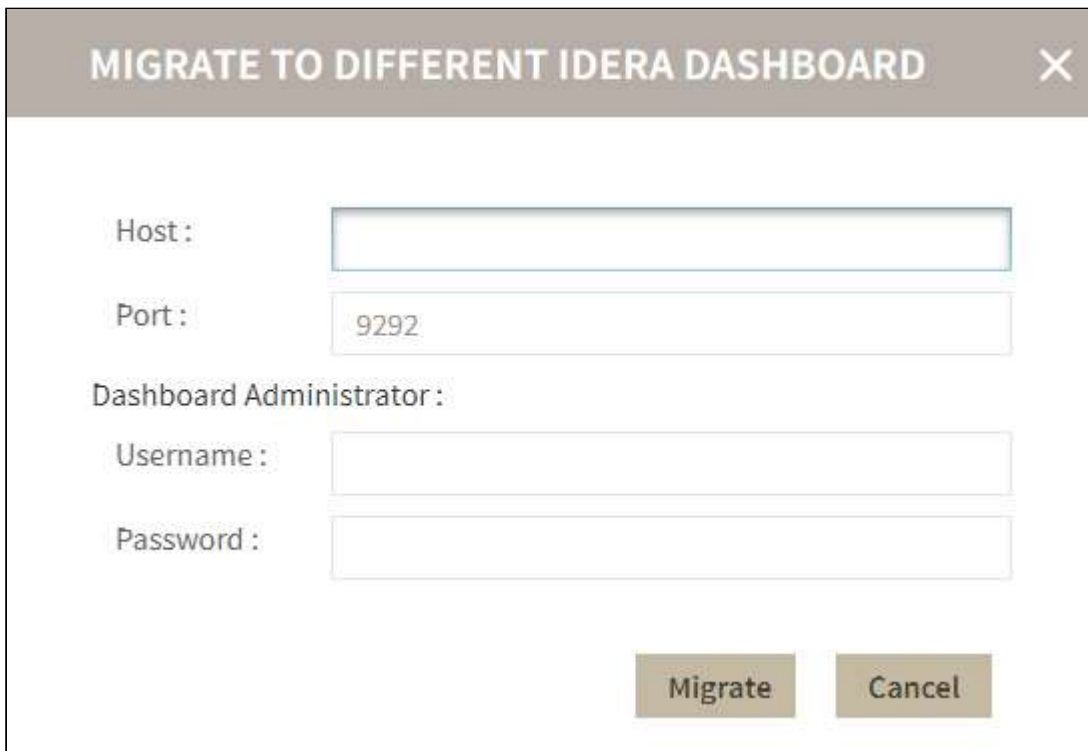
Delete a Product

To remove a product, select it from the Products list and click **Delete** on the panel to the right. A confirmation dialog displays; click **Yes** to remove or **No** to cancel.

⚠ If you want to uninstall the IDERA Dashboard, make sure to delete all registered products before uninstalling.

Migrate Product

Allows you to migrate the registered product to a different IDERA Dashboard. Select the product from the Products list and click **Migrate product**.



MIGRATE TO DIFFERENT IDERA DASHBOARD X

Host :

Port :

Dashboard Administrator :

Username :

Password :

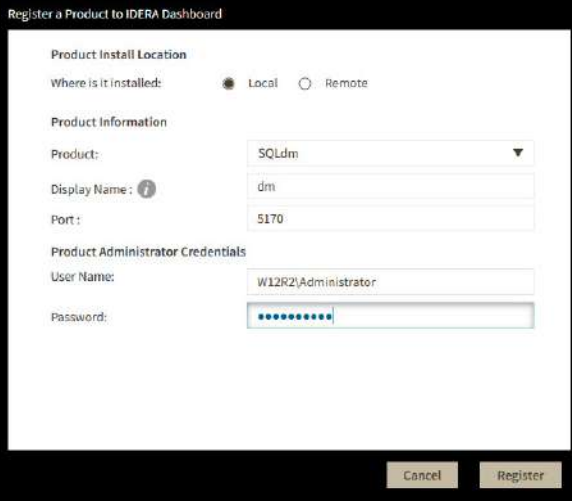
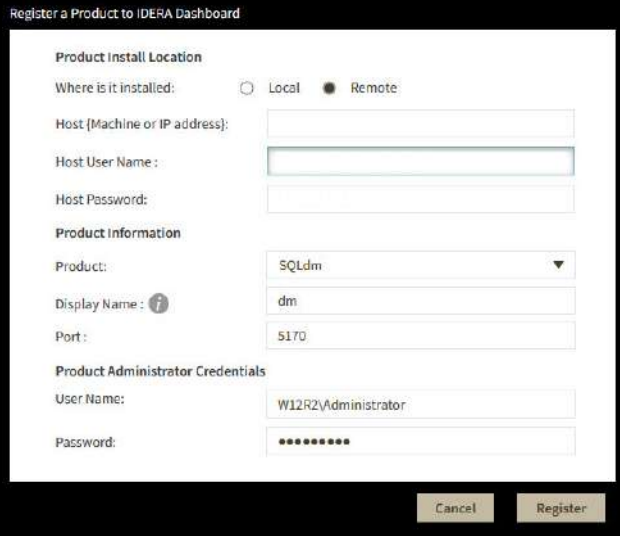

Migrate Cancel

Fill in the requested information and click **Migrate**.

Register SQL Diagnostic Manager

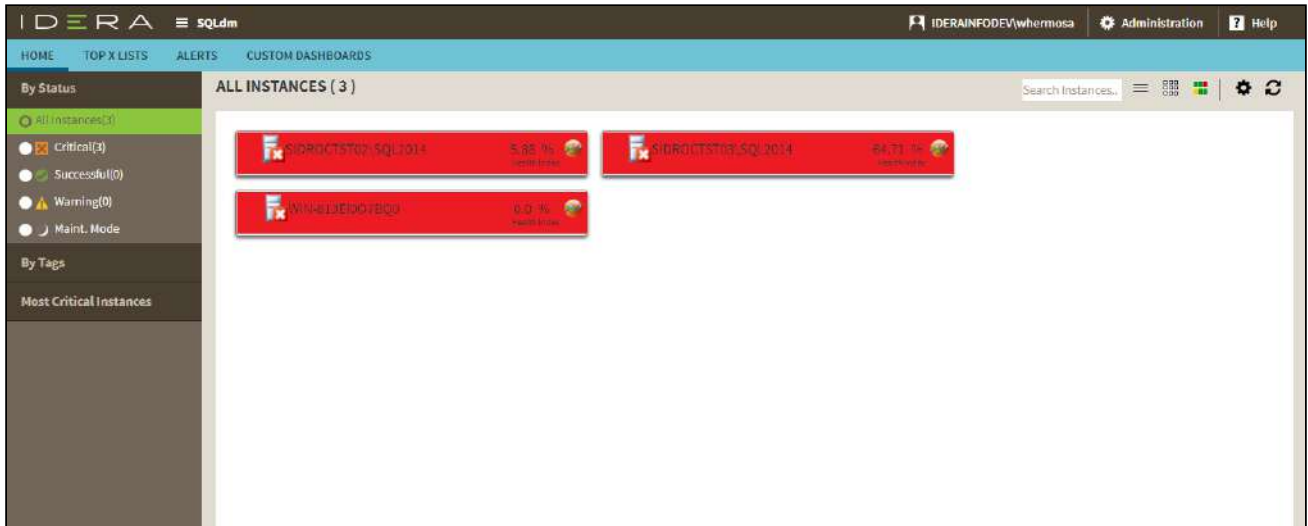
Register SQL Diagnostic Manager in your IDERA Dashboard. Click **Register a Product** to display the configuration settings.

When the Register a Product to IDERA Dashboard window displays, first select the product installation location (Local or Remote) and fill out the information as follows:

Local	Remote
<ol style="list-style-type: none"> 1. In the Product field enter the product name, SQLdm (As it shows in the image below). 2. In the Display Name field enter any name you want. Choose a unique display name to help you distinguish different instances of the same product. 3. Enter the Port number used by SQLdm Rest Service. By default, the port number is 5170. 4. Under Product Administrator enter your credentials. 5. Type all the requested information and click Register. 	<ol style="list-style-type: none"> 1. Enter the Host name or IP Address. 2. In the Host User Name and Host Password, enter the requested information. 3. In the Product Information, type the Product name: SQLdm. 4. In the Display Name field enter any name you want. Choose a unique display name to help you distinguish different instances of the same product. 5. Enter the Port number used by SQLdm Rest Service. By default, the port number is 5170. 6. Under Product Administrator enter your credentials, and click Register.
	
<p> SQL Diagnostic Manager is unregistered from the IDERA Dashboard when you uninstall it.</p>	

Navigate the web console dashboard

The SQLDM web console interface boasts an easy-to-navigate dashboard comprised of several views, sub-views, tabs, and widgets that provide a complete synopsis of your SQL Diagnostic Manager environment. Click the image to view full size.



On the dashboard you can find the following sections:


- [Views by Status, Tags, or Critical Instances](#)
- [List, thumbnail and heatmap sub-views](#)
- [Top X Lists tab](#)
- [Alerts tab](#)
- [Custom Dashboards tab](#)

Additional options

Search

In the SQL Diagnostic Manager web console dashboard, you can search for a particular instance by name. Use the percent (%) wildcard character if you do not remember the full name of the instance that you are looking for.

Heatmap Calculation Configuration

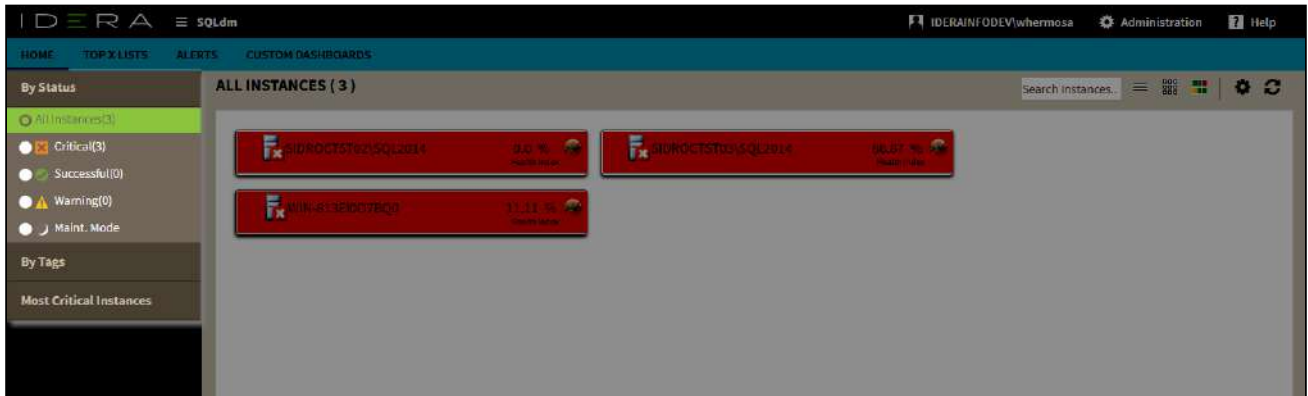
The  icon, located next to the search instances option, allows you to configure the scale factors to calculate the health index displayed in the heatmap.

Refresh

The Refresh button allows you to manually refresh displayed data.

Use views to get an overview of your environment

In the Views section of the home tab you can see at a glance if there are any problems in your monitored SQL Server instances and drill-down for details. Each view is expandable and information is grouped [By Status](#), [By Tags](#), or by [Most Critical Instances](#). SQL Diagnostic Manager displays this information in the panel to the right.



By Status

In the By Status view you can easily identify the status of your monitored SQL Server instances and take action.

Instances are grouped as follows:

- All Instances
-  Critical
-  Successful
-  Warning
-  Maintenance Mode

If you select all instances with an **OK** status in the By Status view, the panel displays information only pertaining to those instances.

By Tags

In the By Tags view, you can view the status of different tags or group of instances, and drill-down to details. For additional information on tags, see [Work with tags](#).

By Most Critical Instances

The Most Critical Instances view lists all the critical SQL servers in your environment and provides links to the single instance dashboard so you can quickly review the issue. The instance with the largest number of alerts shows up at the top of the list.

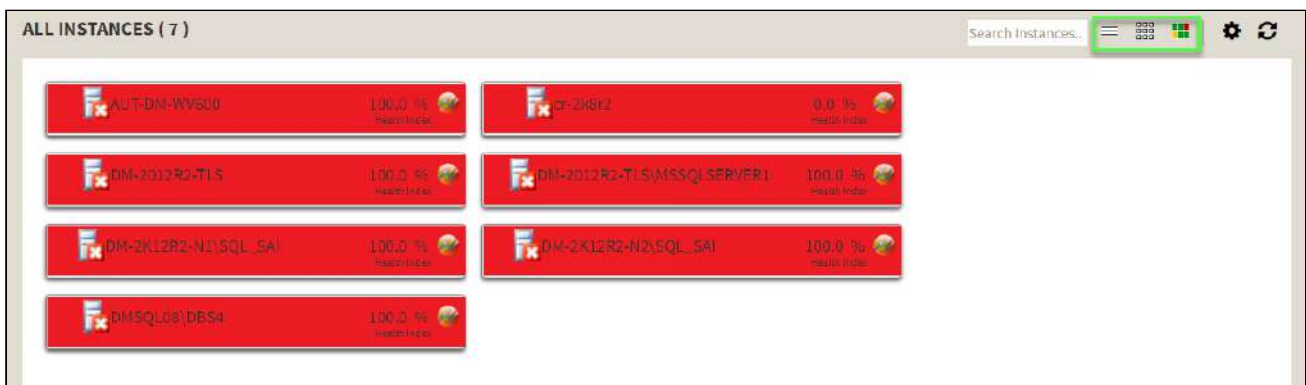
If you click any of the listed critical instances, the web console displays the [single instance dashboard](#).

Use sub-views to identify and diagnose issues

The web console's main dashboard is comprised of three main sub-views:

- [List sub-view](#)
- [Thumbnail sub-view](#)
- [Heatmap sub-view](#)

These sub-views display all data collected at the last refresh for all your monitored SQL Server instances. To access any sub-view, click the appropriate button in the upper-right corner of the dashboard as shown in the following image:



List sub-view

The list sub-view offers a means to quickly identify the status of instances, as well as the possibility to navigate to the single instance dashboard when clicking on a specific row.

SWA	Instance Name	Health	Alert	CPU	Mem...	I/O	Dat...	Logs	Qu...	Version	Status	Agent Status	DTC Status	Availat
	WIN-S13F1007B0Q	0.00%	✖	✖	⚠	✔	✔	✔	✔	14.0.1000.L...	Running	Running	Running	521912
	SI-DROCTST02\SQL2014	37.50%	✖	✔	✔	✔	✔	✔	✔	12.0.2000.8	Running	UnableToM...	UnableToM...	0
	SI-DROCTST03\SQL2014	37.50%	✖	✔	✔	✔	✔	✔	✔	12.0.2000.8	Running	UnableToM...	UnableToM...	0

The list sub-view provides you with the following SQL Server information:

SWA

Displays the SWA icon when the instance is registered in SQL Workload Analysis.

Instance Name

Displays the name of the monitored SQL Server instance.

Health

Displays the health index in percentage in a red square.

Alert

Displays the current severity of the value of the instance.

CPU, Memory, I/O, Database, Logs, and Queries

Displays the overall health index icon for each option.

SQL Server Version

Displays the edition and version of SQL Server software used in this instance.

SQL Server Status

Displays the status of the SQL Server service such as running, stopped, and paused.

SQL Agent Status

Displays the status of the SQL Server Agent service such as running or stopped.

DTC Status

Displays the status of the Distributed Transaction Coordinator service such as running or stopped.

Available Memory (MB)

Indicates how much available memory is allocated for SQL Server.

Blocked Sessions

Displays blocked process sessions information on the SQL Server instance.

CPU Activity

Displays the percentage of CPU consumed by your SQL Server, virtual machine, and host server.

Disk Queue Length

Displays the average number of system requests that are waiting for disk access on the computer hosting the SQL Server instance

Actions

Sort


Instances on the list sub-view are sorted by severity. To re-organize information, click a column header or select the option to group by severity, tags, or SQLdmRepo (**Group By**).

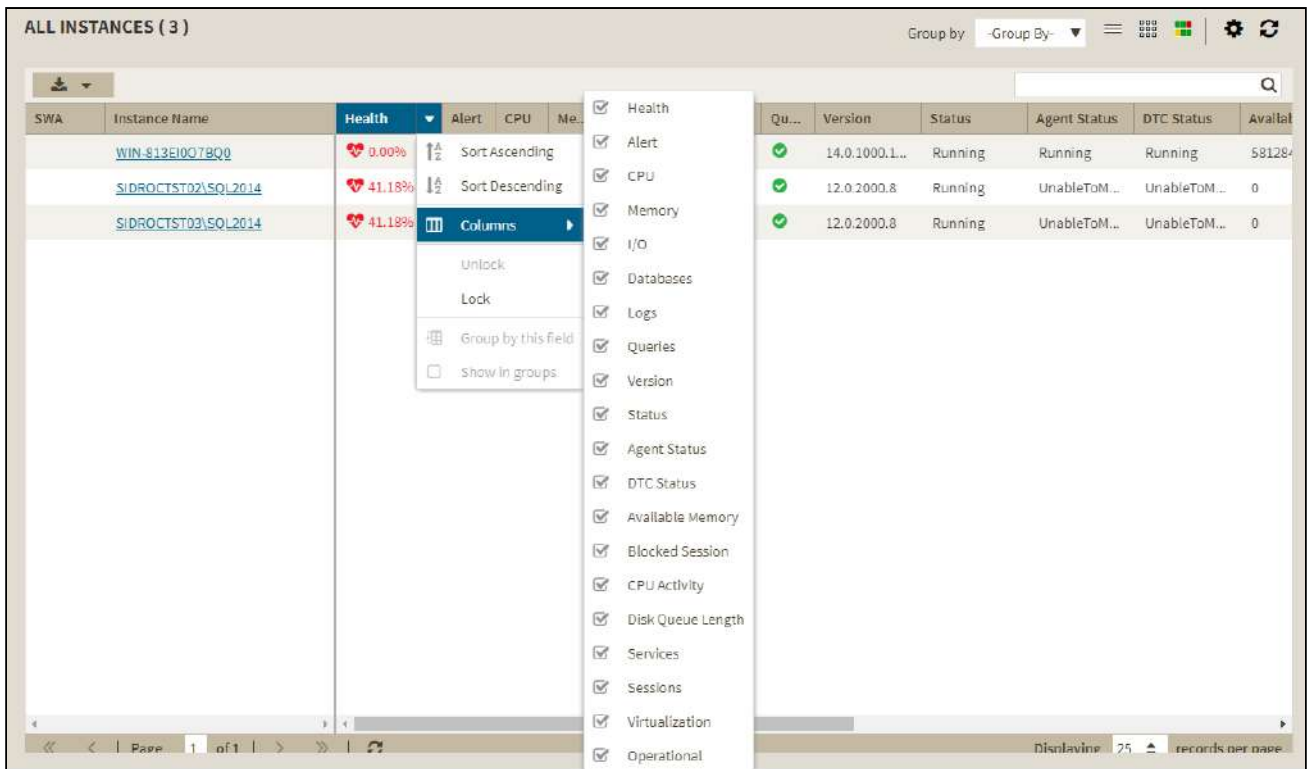
Search Instance

Type the name of the instance you are looking for.

Options

Choose the columns to display in the list view.

The Dashboard console has a new alert grid and custom filters that help you organize the list columns in your most convenient way. Click the  icon located next to the column name to expand the drop down menu. This menu allows you to sort your criteria by Sort Ascending or Sort Descending, you can remove or add columns, lock and unlock, and group the information by fields.




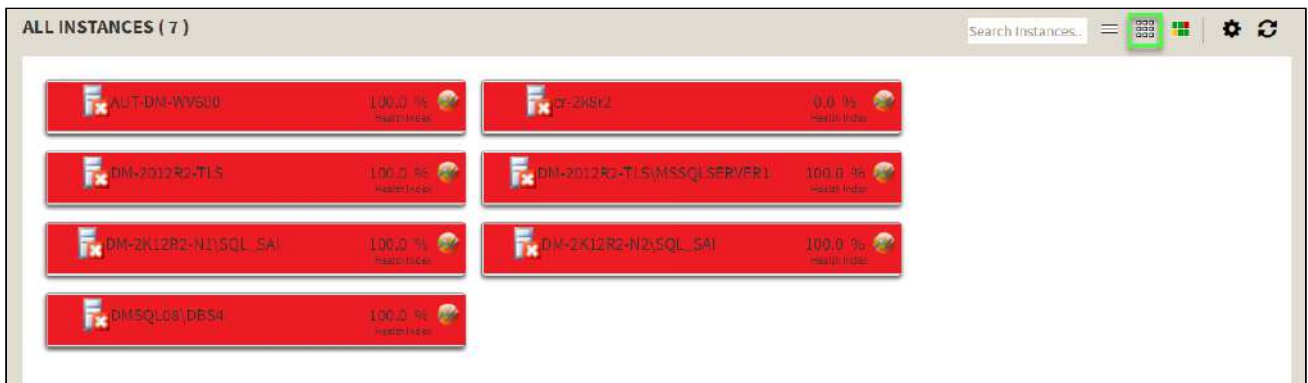
The screenshot shows the 'ALL INSTANCES (3)' dashboard. A table lists instances with columns for SWA, Instance Name, Health, Alert, CPU, and Memory. The 'Health' column is selected, and a context menu is open, showing options like 'Sort Ascending', 'Sort Descending', 'Columns', 'Unlock', 'Lock', 'Group by this field', and 'show in groups'. A secondary menu is also visible, listing various metrics such as Health, Alert, CPU, Memory, I/O, Databases, Logs, Queries, Version, Status, Agent Status, DTC Status, Available Memory, Blocked Session, CPU Activity, Disk Queue Length, Services, Sessions, Virtualization, and Operational.

SWA	Instance Name	Health	Alert	CPU	Me...
	WIN-813E1007BQ0	0.00%		Sort Ascending	
	SIDROCTST02\SQL2014	41.18%		Sort Descending	
	SIDROCTST03\SQL2014	41.18%			

Qu...	Version	Status	Agent Status	DTC Status	Availab
	14.0.1000.1...	Running	Running	Running	581284
	12.0.2000.8	Running	UnableToM...	UnableToM...	0
	12.0.2000.8	Running	UnableToM...	UnableToM...	0

Thumbnail sub-view

The thumbnail sub-view is the default view in the SQL Diagnostic Manager web console's dashboard and provides you with a graphical replication of your monitored SQL servers. Access additional details in [the single instance dashboard](#) by clicking on a thumbnail's instance name or easily spot categories that have active alerts as marked by a critical icon. Navigate to an instance's specific category view in case it has active alerts by clicking on the marked category icon. You can also launch the SQL Diagnostic Manager console directly from the thumbnail sub-view by clicking the SQL Diagnostic Manager icon .



The thumbnail sub-view displays basic statistics about your SQL Servers:

SQL Server version

Displays the edition and version of SQL Server software used in this instance.

Response Time

Refers to the time it takes for a query to go from the SQL Diagnostic Manager Console to the SQL Server instance and back.

User Sessions

Refers to the total number of SQL Server user sessions.

SQL CPU Usage

Refers to average percentage of SQL Server processor usage on the computer hosting the SQL Server instance.

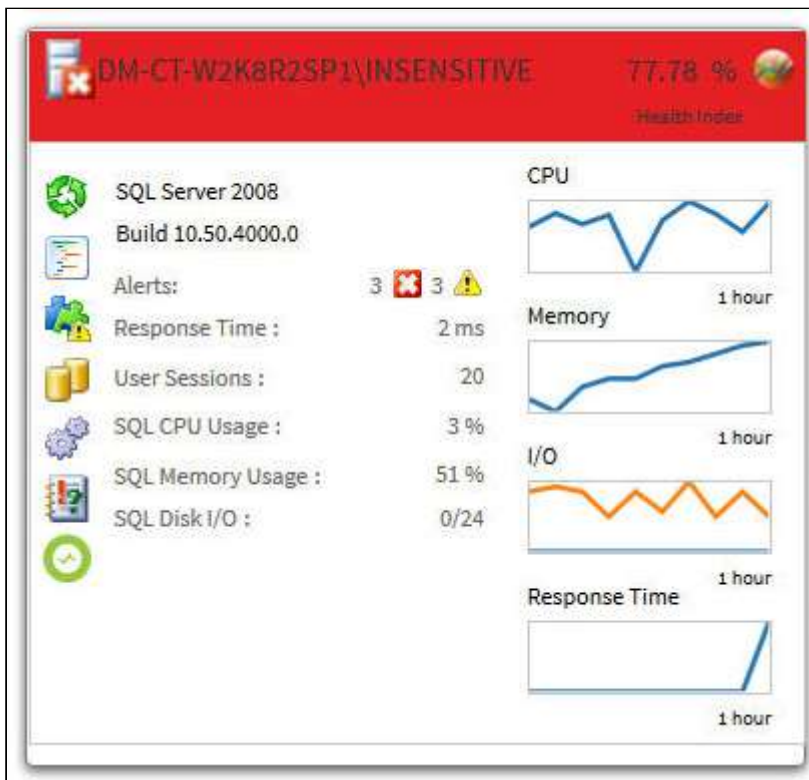
SQL Memory Usage

Refers to amount of memory in use by the monitored SQL Server instance.

SQL Disk I/O

Refers to the number of physical reads and physical writes made by the SQL Server instance between refreshes.






In addition, the thumbnail sub-view includes CPU, Memory, I/O, and Response Time sparkline graphs for a 1 hour-view so that users can obtain an overview of key metrics quickly.






Category icons and actions

In the thumbnail sub-view, use your mouse cursor to hover on a category icon in the left of the thumbnail. If the category icon contains active alerts, you can click it to access the [Alerts Detail screen](#).

If there are no active alerts, you can click a category icon and you are directed to a specific category view in the single instance dashboard. The table below shows the relationship between the category icons and their corresponding views:

Icon	Category
	Sessions tab of the single instance dashboard
	Queries tab of the single instance dashboard
	Resources tab of the single instance dashboard
	Databases tab of the single instance dashboard
	Alerts Detail screen

Icon	Category
	Overview tab of the single instance dashboard
	Launch SWA

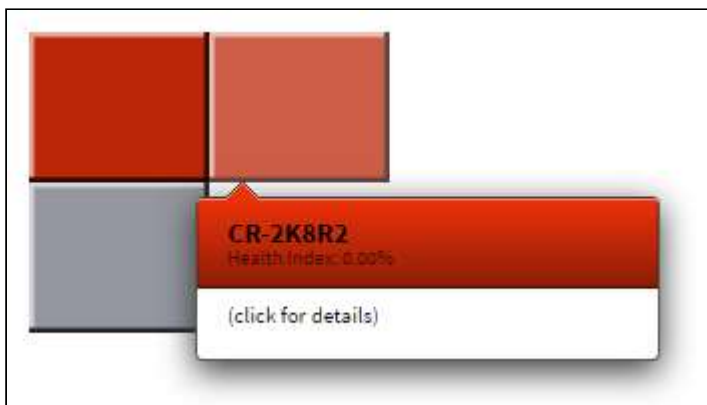
 Instances in thumbnail sub-view display according to severity. This means that critical instances show up on the top of the dashboard.

Heatmap sub-view

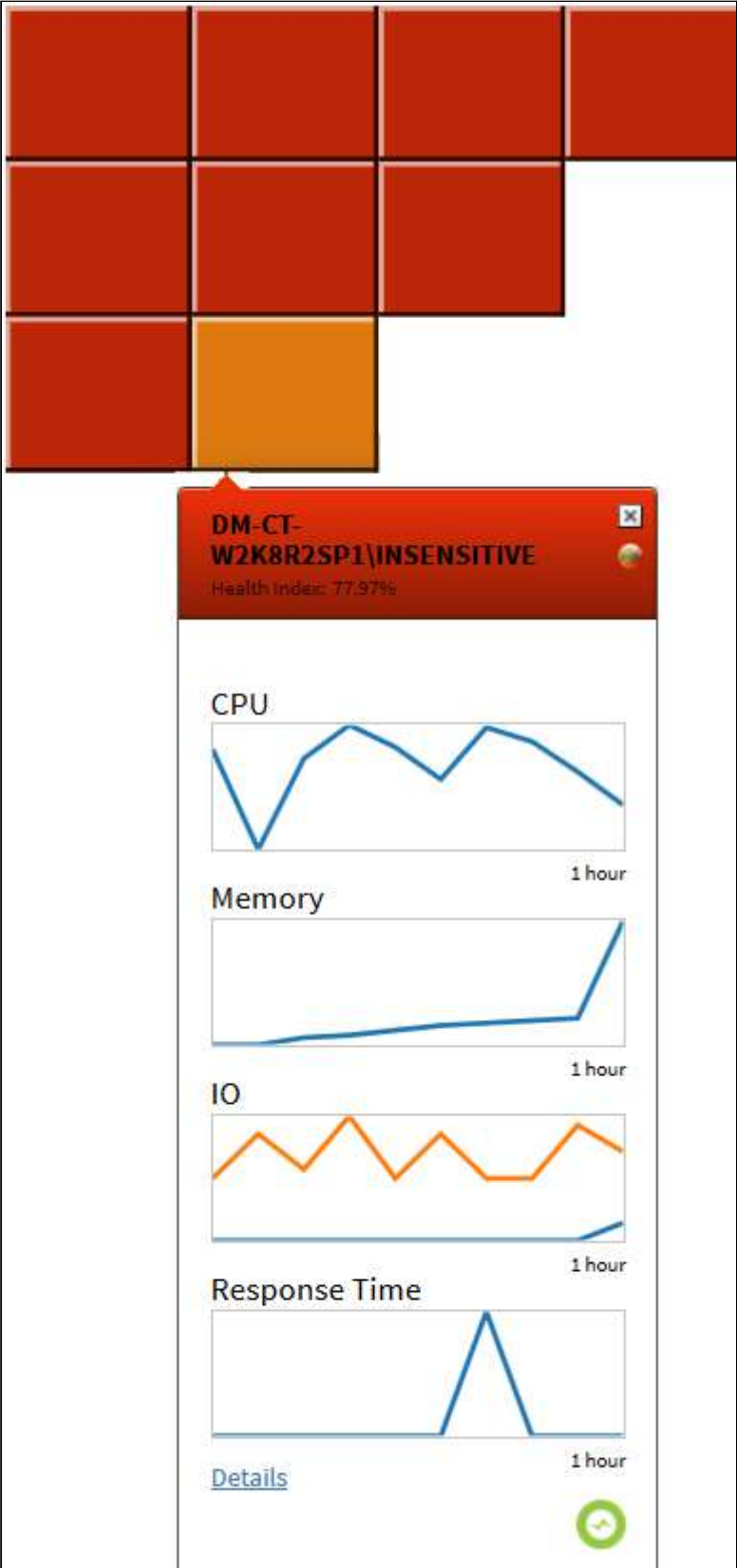
The heatmap sub-view provides you with a visual overall status of your SQL Server environment in the form of a grid of color coded squares that contain a health index. Through its color scheme and health index you can easily identify which of your monitored instances need the most attention. Each square in the heatmap sub-view represents a monitored instance. From each square, you can access basic statistics but also drill down for details.





To glance at instance health information just hover over a specific square to see its health index.



Upon clicking a specific square, the health index displays performance line charts for CPU, Memory, IO, and Response Time from the past hour.



Additionally, you can:

- Click **Details** to access the Overview tab of the single instance dashboard.
- Launch the SQL Diagnostic Manager console by clicking the SQL Diagnostic Manager icon .
- Launch SQL Workload Analysis by clicking the SWA icon .


How to interpret the health index in the heat map sub-view?

The health index is a percentage value that provides an estimate of the relative health of your monitored SQL Server. The health index is based on the following raw calculation:

Health Index Raw Calculation = Instance or Tag Scale Factor * (# Critical Alerts * Critical Scale Factor + # Warning Alerts * Warning Scale Factor + # Informational Alerts * Informational Scale Factor)

The raw score is normalized to a range between 100% and 0%. SQL Server is healthier the closer its health index is to 100%, and it is in critical state the closer its health index is to 0%. Use the severity factors to change the relative weight in the calculation. Use tag and instance factors to assign instances more or less importance in the overall heat map ranking.

To configure the Health Index Calculations:

1. Click the gear icon  at the upper-right corner of the **All Instances** section to display the Heatmap Calculation Configuration dialog.

Heatmap Calculation Configuration

The scale factors specified below are used to calculate the health index displayed in the **heatmap**. The total number of active alerts per severity are multiplied by their respective factor and added to get a raw score which is normalized to a range between 0% and 100%. Use the severity factors to change the relative weight in the calculation. Use tag and instance factors to assign instances more or less importance in the overall heatmap ranking. IDERA recommends factor values between 0 and 10.

Alerts	Scale Factors
Critical	<input type="text" value="6.0"/>
Warning	<input type="text" value="2.0"/>
Informational	<input type="text" value="1.0"/>

Instances	Scale Factors	Disable
<input type="text" value="Choose"/> ▼	<input type="text" value="1"/> <input type="button" value="Add"/>	

Tags	Scale Factors	Disable
<input type="text" value="Choose"/> ▼	<input type="text" value="1"/> <input type="button" value="Add"/>	

- Type the values that you want in the respective Alert Scale Factor field. IDERA recommends factor values between zero and ten.
- Choose an instance from the registered instances and click **Add** to apply a distinctive scale factor.
- Choose a tag from the available tags and click **Add** to apply a distinctive scale factor.
- Click **Submit**.

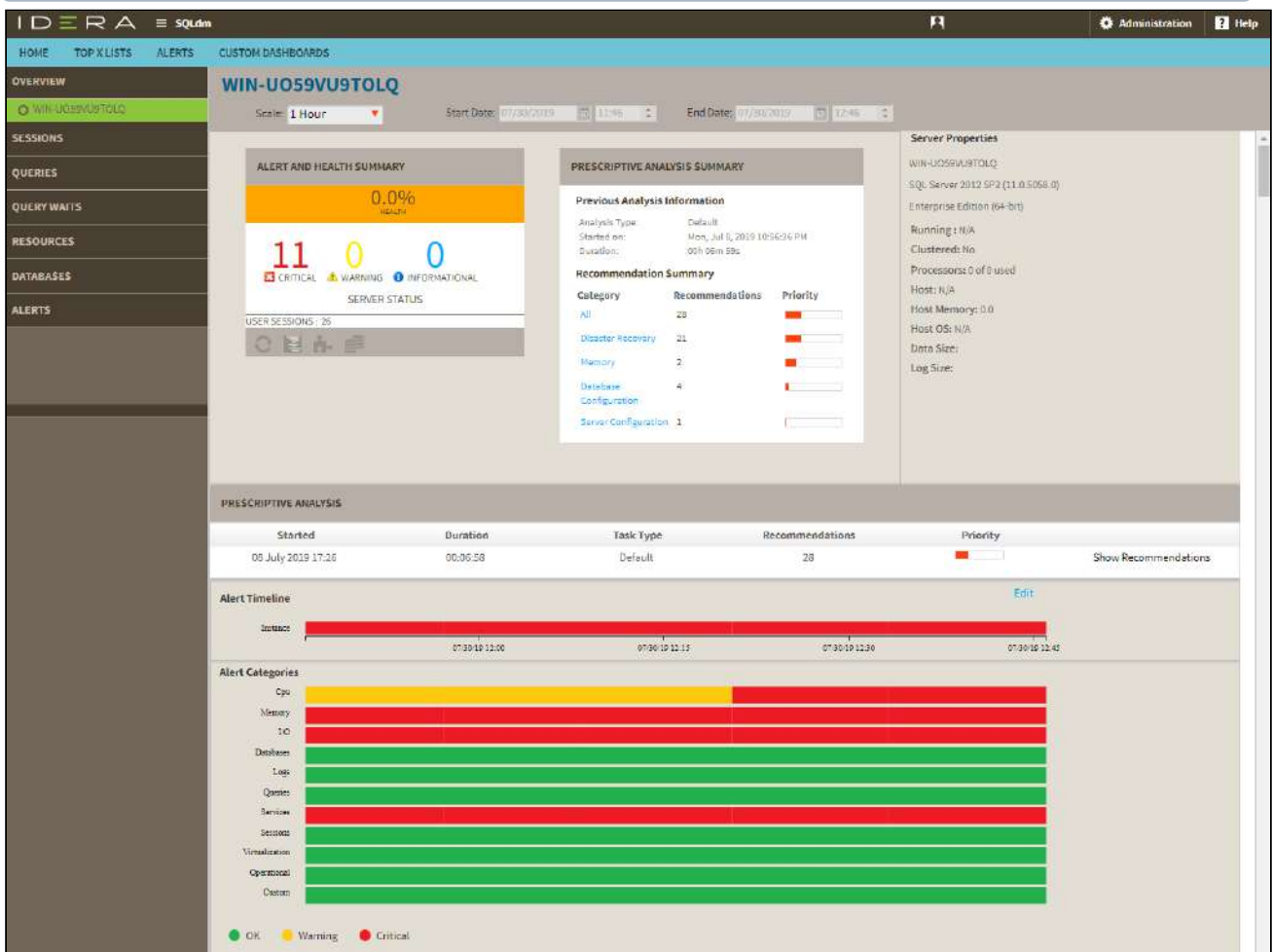
i The heatmap sub-view displays instances according to severity. The color of the square relates to the percentage of its health status and includes a range of colors not only restricted to green, yellow, and red.

Navigate the web console single instance dashboard

The single instance dashboard includes an overview section for the selected monitored SQL server and specific category views. SQL Diagnostic Manager users are able to obtain details on sessions, queries, resources, and databases. Click image to view full size.

SQL Workload Analysis

When the monitored instance it is registered in SQL Workload Analysis, the Launch SWA icon displays on the right side of the History Range Control. Click the Launch SWA icon to be redirected to SQL Workload Analysis.



In the single instance dashboard, you can find the following sections:

- [History Range Control](#)
- [Alert and Health Summary](#)
- [Prescriptive Analysis Summary](#)
- [Server properties](#)
- [Prescriptive Analysis](#)

- [Alert Timeline](#)
- [Alert Categories](#)
- [Charts](#)

History Range Control

SQL Diagnostic Manager also includes a new [History Range Control](#), located at the top of the Web Console and in the option History Browser in the Desktop Console, both can be modified and be designated a specific start time and date as well as current snapshot.

Alert and Health Summary

In this section, you find the health index, the alert status, and the user sessions from SQL Diagnostic Manager. Also, includes shortcuts to filter data through Sessions, Queries, Resources, and Databases.

Prescriptive Analysis Summary

The prescriptive summary shows the previous analysis information and the resulting recommendations summary. You can click any category under Recommendation Summary to be redirected to the Analyze tab in the desktop console

Server Properties

In this section view the most important characteristics of your monitored SQL Server instance.

Instance Name

Displays the name of the monitored SQL Server instance or its friendly name assigned in the desktop console.

- ✓ [Assign a friendly name](#) to your instance that is meaningful to you and easy to identify.

SQL Version

Indicates the current SQL Server version including service pack and build information on this SQL Server instance.

SQL Edition

Indicates the current SQL Server edition, including Enterprise, Business Intelligence, or Standard, installed on this SQL Server instance.

Running since

Indicates the timestamp associated with the most recent restart of this SQL Server instance.

Clustered

Indicates whether this SQL Server instance is part of a cluster. If it is part of a cluster, SQL Diagnostic Manager displays the name of the active node.

Processors

Indicates the total number of processors associated with this SQL Server instance and how many of these processors are currently in use.

Host

Indicates the name of the server hosting this SQL Server instance.

Host Memory (GB)

Indicates the total amount of memory on the server hosting this SQL Server instance.

Host OS

Indicates the current operating system on the server hosting this SQL Server instance.

Data Size

Displays the data size of the server.

Log Size

Displays the log size of the server.

Prescriptive Analysis

The Prescriptive Analysis panel shows a summary of the performed Analyses, this panel shows the Date, the Duration, the Task Type, the number of recommendations, the Priority, and you can click the **Show Recommendations** to review the list of recommendations of each Analysis.

Alert Timeline

This graph shows a summary of the alert categories, CPU, Memory, I/O, Databases, Logs, Queries, Services, Sessions, Virtualization, Operational and Custom. This graph is color-coded depending on the status of the alerts inside each category.

Alert Categories

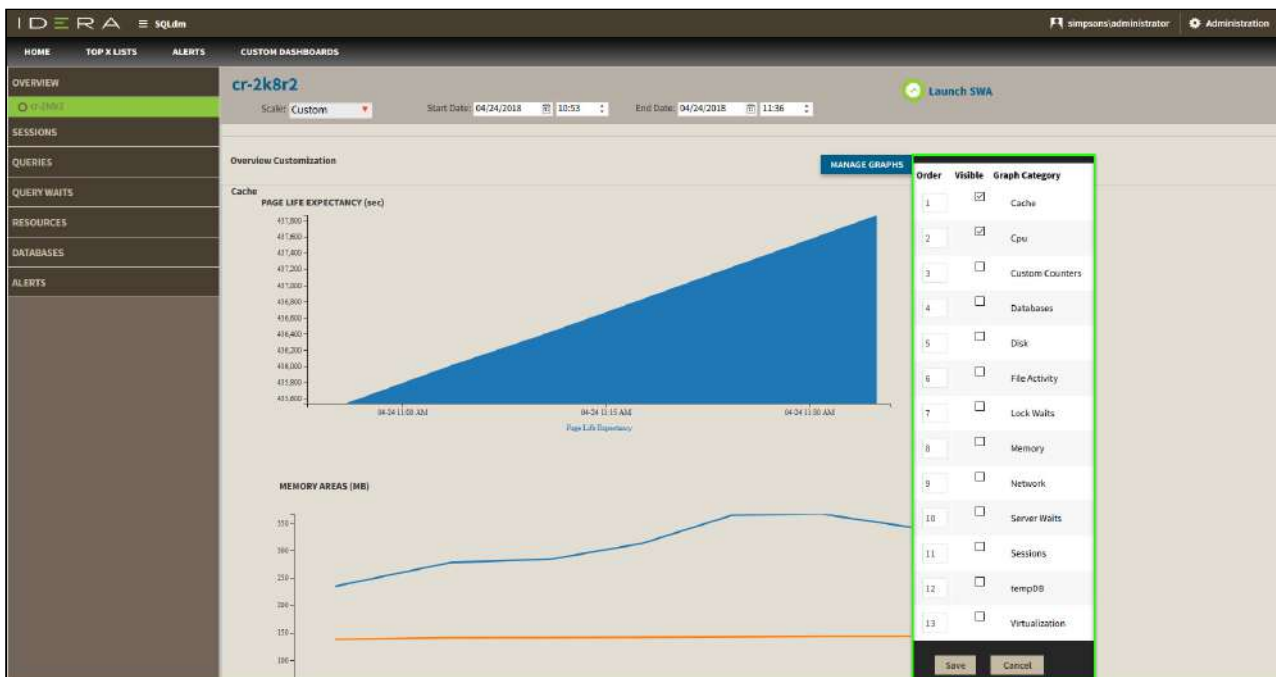
This alerts timeline view allows you to easily identify the highest alert for your monitored SQL Server instance and the highest alert for a specific category.

Configure the Alert Category graph, following these steps:

- Click **Edit**, located next to the Alert Timeline graph.
- Check or uncheck the check boxes located under the Visible column.
- Or, reorganize the Order column with the listed numbers preferred.
- Click **Save**.

Charts

The single instance dashboard includes the option **MANAGE GRAPHS**. This option allows you to choose the following graphs to be displayed in the overview single instance view:



Cache

The Cache chart tracks the database and procedure processes using a buffer pool of memory on your monitored SQL Server instance.

CPU

The CPU chart tracks the performance of the processor used by your monitored SQL Server.

Custom Counters

The Custom Counters chart displays the statistics for a selected custom counter on a monitored SQL Server instance.

Databases

The Databases chart tracks the performance of the databases on your monitored SQL Server.

Disk

The Disk chart tracks the performance of the disk used by your monitored SQL Server.

File Activity

The File Activity chart displays the top five database files with the highest relative activity since the last refresh.

Lock Waits

The Lock Waits chart tracks the total number waits caused when a task is waiting to acquire a lock on your monitored SQL Server.

Memory

The Memory chart tracks the usage and availability of memory on your monitored SQL Server computer.

Network

The Network chart tracks the performance of the network connection being used by your monitored SQL Server.

Server Waits

The Server Waits chart tracks the total number and time spent on waits affecting your monitored SQL Server.

Sessions

The Sessions chart tracks the performance of the active and blocked sessions running on your monitored SQL Server.

tempDB

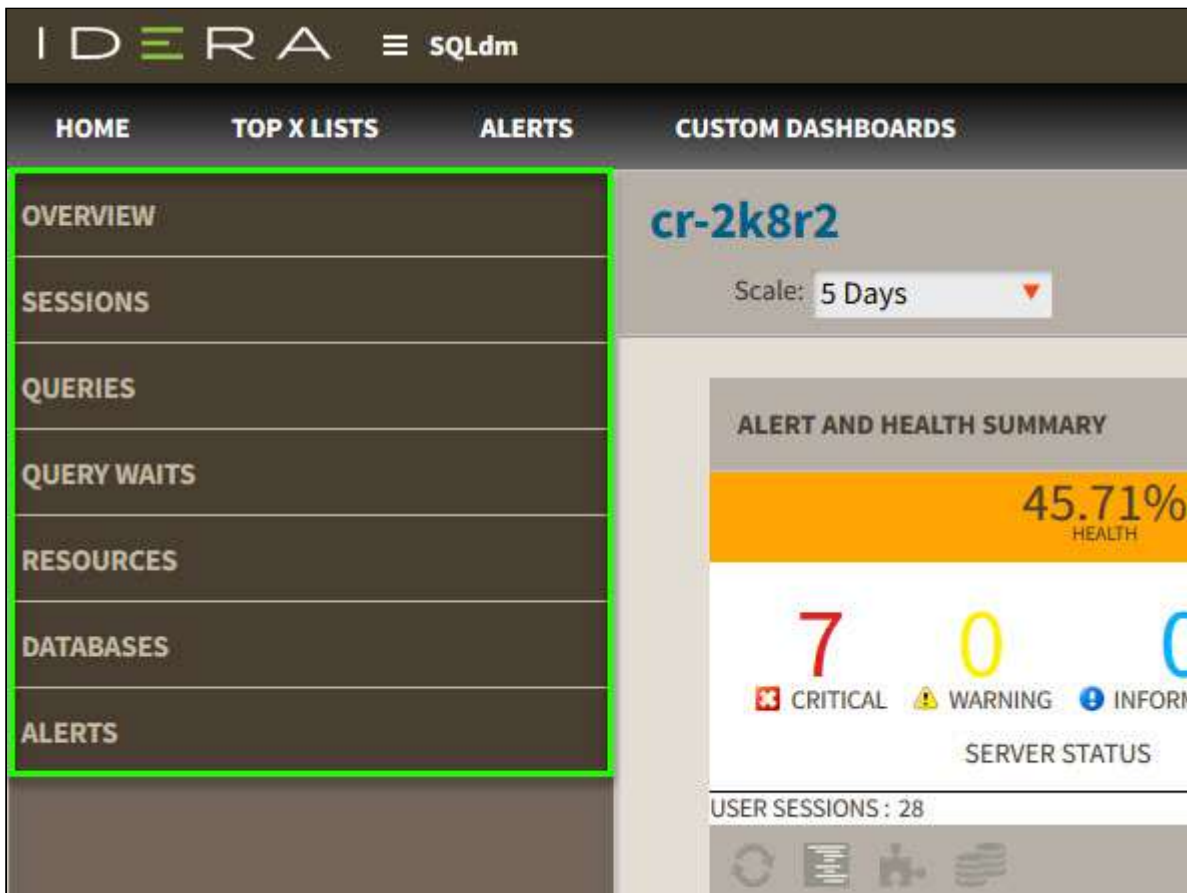
The Tempdb chart tracks the status of the tempdb database on your monitored SQL Server instance.

Virtualization

The Virtualization chart tracks the performance of your virtual machine or host, depending on the selection in the panel.

Category views

The single instance dashboard contains the Overview, Sessions, Queries, Query Waits, Resources, Databases, and Alerts tabs where you can see metrics and information on these category views.



From these tabs you can:

- [View your SQL Server sessions information.](#)
- [View your SQL Server queries information.](#)
- [View your SQL Server query waits information.](#)
- [View your SQL Server resources information.](#)
- [View your SQL Server databases information.](#)
- [View your SQL Server alerts.](#)

Access the single instance dashboard

SQL Diagnostic Manager provides several paths to access the single instance dashboard. The first access path is by clicking the appropriate monitored SQL Server instance in the list, thumbnail, or heatmap sub-views. The second access path is by clicking a specific instance from the Active Alerts panel, or the Most Critical Instances view in the main dashboard. It is also possible to access the single instance dashboard by clicking a specific instance or database in the Top servers by alert count, Top servers by response time, and Top databases by alerts widgets.

View your SQL Server sessions information

The Sessions tab of the single instance dashboard allows you to quickly view key diagnostic statistics for sessions on your SQL Server instance and contains the Overview and Session Graphs views. Click image to view full size.

Session ID	Host	Database	Status	Open Transactions	Command	Application	Wait Time (ms)	Wait Type	Wait Resource	CPU (ms)	CPU Delta (ms)	Physical I/O	Memory
73	CR-2K8R2	IderaDashboardRepository	Runnable	0	UPDATE	Idera Core Services	0	N/A		0	0	0	0
52	CR-2K8R2	msdb	Sleeping	0	N/A	SQLAgent - Generic Refresher	0	N/A		1344	0	41	16
51	CR-2K8R2	msdb	Sleeping	0	N/A	SQLAgent - Job Invocation engine	0	N/A		312	0	45	16

Overview

The **Sessions > Overview** tab contains the following information:

- Session ID
- Host
- Database
- Status
- Open transactions
- Command
- Application
- Wait time (ms)
- Wait type
- Wait resources
- CPU (ms)
- CPU Delta (ms)
- Physical I/O
- Memory
- Login time
- Last Batch
- Blocked by
- Blocking
- Blocking count
- Net address
- Net library
- Version store elapsed (sec)

✓ Export

You can export sessions overview information in the following formats: PDF, XLS, or XML.

Session Graphs

The **Sessions > Sessions Graphs** tab contains the following charts (Click image to view full size):

Response Time (ms)

The Response Time chart displays the response time (in milliseconds) it takes a Select SQL statement to go to the SQL Server instance and back. This chart indicates the current speed and congestion of the network as well as the speed with which SQL Server processes small queries.

Session Activity

The Session Activity chart displays the number of concurrent connections to the monitored SQL Server instance over time. The total number of sessions includes the active, idle, and system sessions.

Blocked Sessions

The Blocked Sessions charts lists all the Blocked Sessions, Lead Blockers, and the Total Deadlocks for the selected SQL Server instance. This allows you to easily view blocks as they occur.

Access the Sessions tab

SQL Diagnostic Manager provides three paths to access the Sessions tab. The first access path is by clicking on the Response Time chart (ms) of the single instance dashboard, and you are directed to the **Sessions > Overview** tab. The second access path is by selecting the Sessions tab of the single instance dashboard. You can also click on an alert that belongs to the Sessions category in the Active alerts panel of the main dashboard.

View your SQL Server queries information

The Queries tab of the single instance dashboard lists each event that occurs in the Query Monitor and allows you to easily explore query data in multiple ways and get a handle of query performance in your environment.

The Queries tab provides a highly usable and customizable Main Query view along with two low-level views which you can navigate to for root cause analysis:

- [Query Signature view](#)
- [Query Details view](#)


The Main Query view consists of the following major components that allow you to customize your view:

- [View selection](#) and [grouping](#)
- [Filtering](#)
- [Time](#)
- [Data view](#)

Select the type of view that you want displayed

The View Selection of the Main Query view allows you to choose from a variety of views to explore query data. The following views are available:


- Duration (ms)
- CPU Time (ms)
- Reads
- Writes
- I/O (sum of reads and writes)
- Blocking Duration (ms)
- Wait Duration (ms)
- Deadlocks

 Duration (ms) is the default view of the Main Query view.

Use Group By to organize your view

Once you have selected a view, use the **Group By** option to choose from the following type of objects for query aggregation:

- Application
- Database
- User
- Client
- Query Signature

 Application is the grouping by default in the Main Query view.

Use Filtering to customize your view


Filter options in the Main Query view allow you to focus only on the information you want to be displayed. You can include or exclude specific applications, databases, users, SQL text, and more from the Query Monitor results through filters. The following filter options are available:


- Application
- Databases
- Users
- Clients
- SQL (Include SQL text matching or exclude SQL text matching)
- [Advanced Filters](#)

Select a time period to display query data

You can choose the time period to display in the Main Query view. This selection limits the queries to only those that run during the specified time period. You can select from a broad range of built-in time periods or specify start and end dates, and times for the period. The **Time** box includes the following options:

- Last 30 minutes
- Last 1 hour
- Last 4 hours
- Last 8 hours
- Last 24 Hours
- Last 7 days
- Last 30 days
- Custom

 By default only queries that start within the time period are shown. If you want to modify this behavior, the Advanced Filters option lets you include incomplete queries or overlapping queries. For more information, see [Advanced filters](#) of the main query view.

 To switch to a shorter time frame, click the larger time bucket. For example, clicking a month in a monthly graph modifies the time period to the selected month and grouping to days.

Analyze your query data

On the right-hand side of the overall view, query data is displayed based on your View, Group By, and filtering selections. The Data view is designed to provide good visualization of the selected data and enable you to drill down into problem areas. The Data view is composed of the following elements:

Header

Provides View, Group By, Period, and filter criteria information.

Graph

Chart area which displays the aggregated query results that meet the filtering criteria.

List

Displays data selected for the view in list form. There are several [list options](#) according to your **Group By** selection in the Main Query view.

The table below shows what options you have to drill down from the graph or list and get more specific information:

Click the...	Access the...
Application Name	Query Signature view
Database Name	Query Signature view
User	Query Signature view
Client	Query Signature view
Query Signature	Query Statement view


You can return to the higher level view by clicking  by the list name.

Access the Queries tab

SQL Diagnostic Manager provides two paths to access the Queries tab. The first access path is by selecting the Queries tab of the single instance dashboard. You can also click the query category icon of a specific instance in the thumbnail sub-view.


Advanced query signature view

SQL Diagnostic Manager provides the ability to view individual SQL Statements or to view Query Signatures. Query signatures are groupings of SQL Statements that match once their literals are stripped. A query signature broadly defines queries and trends with a less overwhelming amount of data and allows you to diagnose a query in a general sense.

Once you identify a potential problem signature, you can drill into individual queries that make up the signature as needed by clicking the  icon.

QUERY SIGNATURE (1) Options Show Top 100 200 All							
Query ID	Signature SQL Text	Occurrences	Event Type	Total Duration (m)	Total CPU Time (ms)	Total Reads	Total Writes
Query Signature 1	select trp.ID Tagl...	1	Stored Procedu	6,149	0	4	0

View SQL
Query Statement

Upon clicking the  icon, the following context menu displays:

- **View SQL.** Click this option to view the underlying SQL Text for the query execution plan. In this view, you can export and save SQL text for review

Query Details ✕

SQL Text
[Export](#)

```

1 -- SQLInventoryManager -- Copyright © 2013-2014-2016, Idera, Inc., All
2   Rights Reserved. set transaction isolation level read
3   uncommitted; set lock_timeout #; set implicit_transactions off;
4   if @@trancount > # commit transaction; set language us_english;
5   set cursor_close_on_commit off; set query_governor_cost_limit
6   #; set numeric_roundabort off; set deadlock_priority low; set
7   nocount on; use master;
8
9   if object_id(#) is not null drop table #db_files;
10
11  create table #db_files( [db_id] int, [file_id] int, [type] tinyint,
12    [page_count] bigint, [pages_used] bigint ) declare @cmd
13    nvarchar(#) declare @db_name nvarchar(#) declare db_cursor
14    cursor static read_only for select name from sys.databases
15    where HAS_DBACCESS(name)=# order by database id open db_cursor
16    fetch next from db_cursor into @db_name while @@FETCH_STATUS =
17    # begin set @cmd = # + replace(@db_name,#,#) + # + # + # + # + #
18    + # begin try insert into #db_files execute (@cmd) end try begin
19    catch -- try sys.master_files to get size but not space used
20    set @cmd = #+@db_name+ # + # + # + # + # + #+@db_name+ # begin try
21    insert into #db_files execute (@cmd) end try begin catch -- add
22    entry so we know about database insert into #db_files values
23    (DB_ID(@db_name), #, #, NULL, NULL) end catch end catch fetch
24    next from db_cursor into @db_name end close db_cursor
25    deallocate db_cursor select d.name, db.* from #db_files db join
26    sys.databases d on d.database_id = db.db_id drop table
  
```


Close

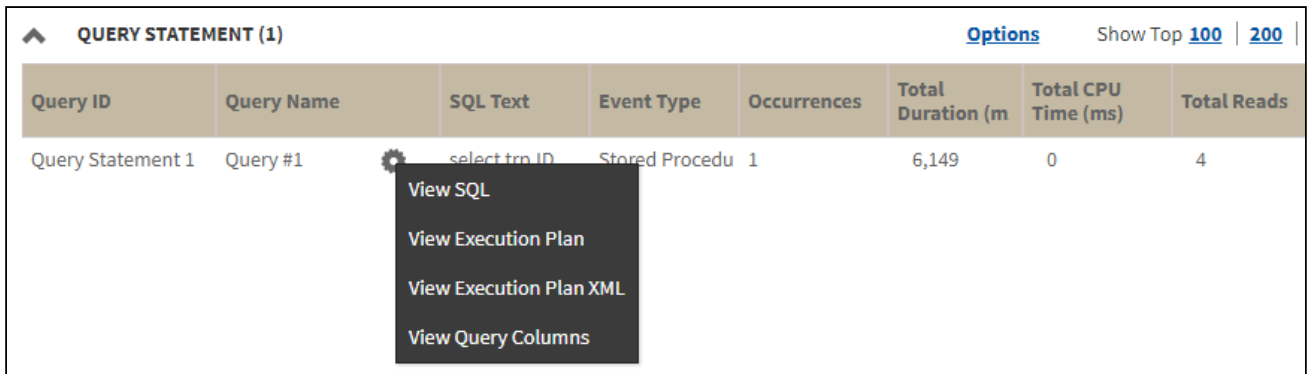
- **Query Statement.** Click this option to view [query statements](#) for the selected client.

For additional information on query signatures, see [View the query monitor signature mode](#).


Advanced query statement view

SQL Diagnostic Manager provides you the ability to view individual SQL Statements or to view Query Signatures. Query statements are presented exactly as they are collected by the Query Monitor Trace. Query statements provide all the detail you may need to diagnose a specific problem with a query.

Once you identify a potential problem statement, you can drill into the Query Details view by clicking the  icon.



Query ID	Query Name	SQL Text	Event Type	Occurrences	Total Duration (m)	Total CPU Time (ms)	Total Reads
Query Statement 1	Query #1	select top ID	Stored Procedu	1	6,149	0	4

Upon clicking the  icon, the following context menu displays:

- [View SQL](#)
- [View Execution Plan](#)
- [View Execution Plan XML](#)
- [View Query Columns](#)

Query Details view

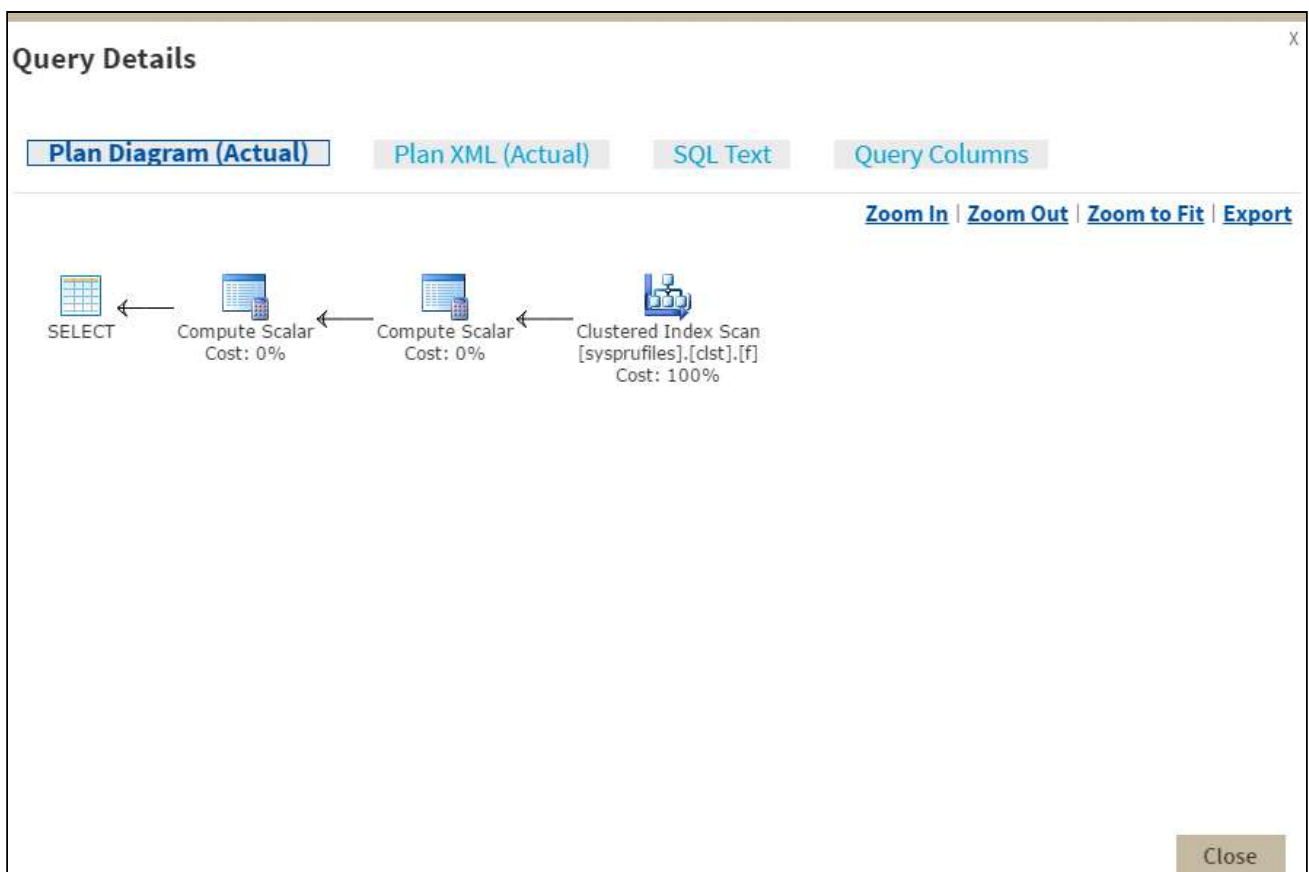
The Query Details view provides various views with detailed information on each query. The Query Details view is comprised of the following tabs:

- [Plan Diagram](#)
- [Plan XML](#)
- [SQL Text](#)
- [Query Columns](#)

⚠ The Plan Diagram and Plan XML views are only available for instances running SQL Server 2008+.

Plan Diagram


The Plan Diagram is a visual representation of the query execution plan (actual or estimated) available in XML format in SQL Server. The Plan Diagram displays a view of the tree of operations that make up a query. This tree shows individual operation nodes, pertaining graphical execution plan icon, along with basic information such as operator name and operation percentage of total cost. Click image to view full size.



In the Plan Diagram view you can perform the following actions:

- Select individual operators and view the [set of properties for each operator](#).
- Increase the size of the image (**Zoom in**).
- Reduce the image size (**Zoom out**).
- Select a size for the view so that fits the current view size (**Zoom to fit**).

- Export image of Plan Diagram.

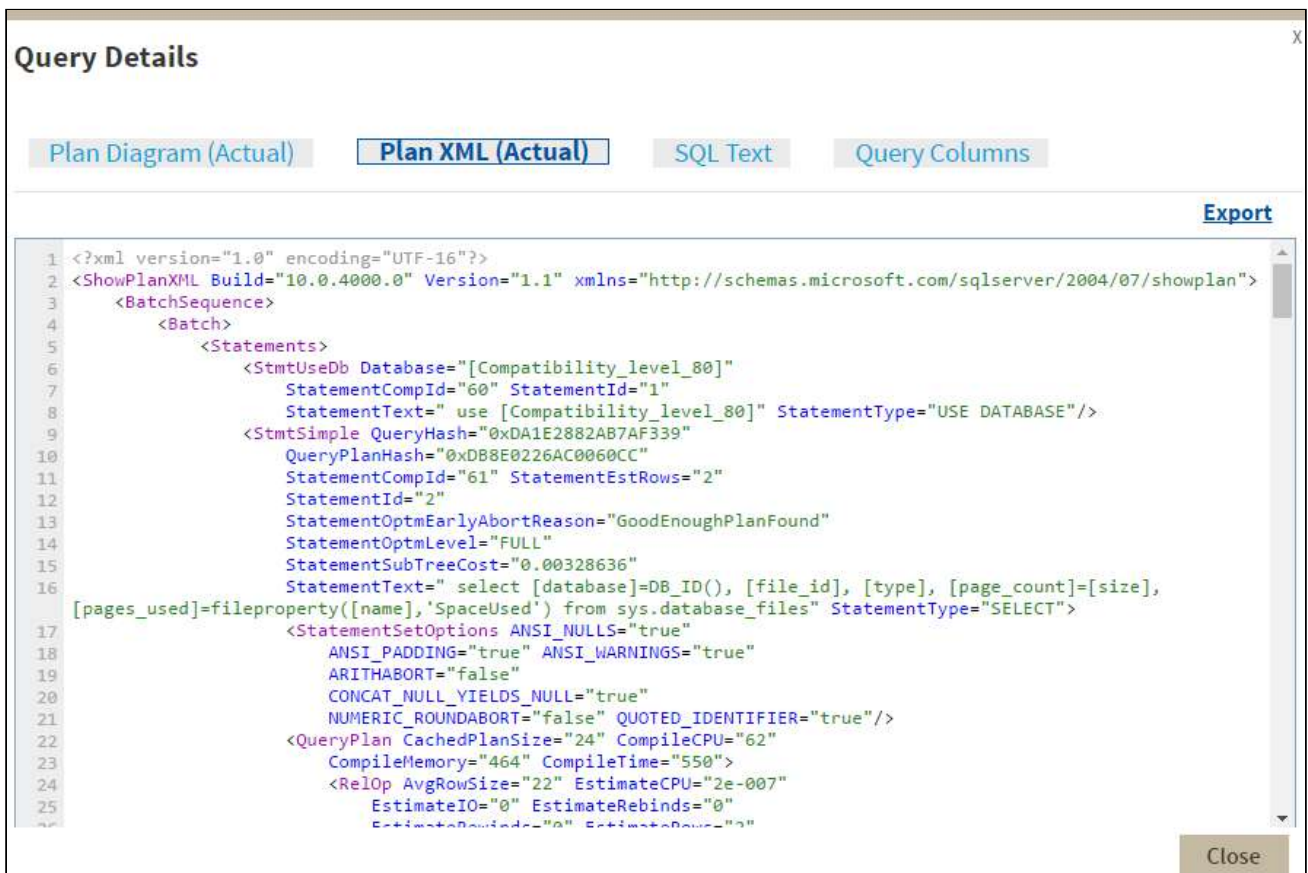
 The top three operators based on the percentage of total cost in a Plan Diagram are highlighted in yellow.

Estimated Query Plans

SQL Diagnostic Manager displays information on whether the plan displayed is an estimated query plan or an actual one.

Plan XML

The Plan XML tab of the Query Details window displays the actual XML of the query execution plan. It has a syntax-highlighting (color-coded) XML viewer. Click image to view full size.



Query Details

Plan Diagram (Actual) | **Plan XML (Actual)** | SQL Text | Query Columns

[Export](#)

```


1 <?xml version="1.0" encoding="UTF-16"?>
2 <ShowPlanXML Build="10.0.4000.0" Version="1.1" xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showplan">
3   <BatchSequence>
4     <Batch>
5       <Statements>
6         <StmtUseDb Database="[Compatibility_level_80]"
7           StatementCompId="60" StatementId="1"
8           StatementText=" use [Compatibility_level_80]" StatementType="USE DATABASE"/>
9         <StmtSimple QueryHash="0xDA1E2882AB7AF339"
10          QueryPlanHash="0xDB8E0226AC0060CC"
11          StatementCompId="61" StatementEstRows="2"
12          StatementId="2"
13          StatementOptmEarlyAbortReason="GoodEnoughPlanFound"
14          StatementOptmLevel="FULL"
15          StatementSubTreeCost="0.00328636"
16          StatementText=" select [database]=DB_ID(), [file_id], [type], [page_count]=[size],
17 [pages_used]=fileproperty([name], 'SpaceUsed') from sys.database_files" StatementType="SELECT">
18           <StatementSetOptions ANSI_NULLS="true"
19             ANSI_PADDING="true" ANSI_WARNINGS="true"
20             ARITHABORT="false"
21             CONCAT_NULL_YIELDS_NULL="true"
22             NUMERIC_ROUNDABORT="false" QUOTED_IDENTIFIER="true"/>
23           <QueryPlan CachedPlanSize="24" CompileCPU="62"
24             CompileMemory="464" CompileTime="550">
25             <RelOp AvgRowSize="22" EstimateCPU="2e-007"
26               EstimateIO="0" EstimateRebinds="0"
27               EstimateRows="2" EstimateRows="2"

```

[Close](#)

In the Plan XML view you can perform the following actions:

- Export and save XML file for review in other application such as SSMS or email query execution plan to other DBAs.

 Save the file with **.sqlplan** extension to view the graphical query plan in SSMS.

SQL Text

The SQL Text tab of the Query Details window shows the underlying SQL Text for the query execution plan.



In the SQL Text view you can perform the following actions:

- Export and save SQL text for review.
- Access the Plan Diagram, Plan XML, and Query Columns tabs.

Query Columns

The Query Columns tab of the Query Details window shows all the referenced columns for the query execution plan.

Query Details X

Plan Diagram (Actual) Plan XML (Actual) SQL Text **Query Columns**

[Export](#)

Database	Schema	Table	Column
[Compatibility_level_80]	[sys]	[sysprufiles]	fileid
[Compatibility_level_80]	[sys]	[sysprufiles]	filetype
N/A	N/A	N/A	Derived Column
N/A	N/A	N/A	Derived Column
N/A	N/A	N/A	Derived Column
N/A	N/A	N/A	Derived Column
N/A	N/A	N/A	Derived Column
N/A	N/A	N/A	Derived Column
N/A	N/A	N/A	Derived Column
[Compatibility_level_80]	[sys]	[sysprufiles]	size
N/A	N/A	N/A	Derived Column

[Close](#)

In the Query Columns view you can perform the following actions:

- Export and save information to a text file.

Plan diagram operator properties

In the Plan Diagram view, hover over an operator to view its full set of properties such as:

- Operator Type.
- [Operator Description](#).
- List of Properties (varies according to operator type).
- Related Objects (varies according to operator type).

Query Details

Plan Diagram (Actual) | Plan XML (Actual) | SQL Text | Query Columns

Zoom In | Zoom Out | Zoom to Fit | Export

The tooltip for the Filter operator displays the following properties:







The FILTER operator returns only rows matching a filter.	
Physical Operation	Filter
Logical Operation	Filter
Actual Number of Rows	27
Estimated I/O Cost	0
Estimated CPU Cost	0.000081
Number of Executions	1
Estimated Operator Cost	0.000081 (0%)
Estimated Subtree Cost	0.0001823
Estimated Number of Rows	9,099,445
Estimated Row Size	445 B
Node ID	5
Output List	
[SYSSESSIONS]	session_id
[SYSSESSIONS]	host_process_id
[SYSSESSIONS]	status
[SYSSESSIONS]	cpu_time
[SYSSESSIONS]	memory_usage
Expr1001	
Expr1002	
Expr1003	









Close










Execution plan graph operators table










Operators describe how SQL Server executes a query or a Data Manipulation Language (DML) statement. The query optimizer uses operators to build a query plan to create the result specified in the query, or to perform the operation specified in the DML statement. The query plan is a tree consisting of physical operators.










This is the table of descriptions to be used in the properties dialog showing the operator details in the execution plan graphical view.







Operator	Graphical execution plan icon	Text
Aggregate	None	The Aggregate operator is a physical or logical operator that computes a new value using the SQL Server functions MIN, MAX, SUM, COUNT or AVG.
Arithmetic Expression		The Arithmetic Expression operator computes a new value from existing values in a row. Arithmetic Expression is not used in SQL Server 2014.
Assert		The Assert operator verifies a condition. The Assert operator is a physical operator.
Assign		The Assign operator assigns a value to a variable
Asnyc Concat	None	The Asnyc Concat operator is used in remote distributed queries to get output rows from remote child nodes to send to the parent node.
Bitmap		The Bitmap operator is used to apply filtering to parallel query plans. Bitmap is a physical operator.
Bitmap Create		The Bitmap Create operator is a logical operator that shows where bitmaps are built.
Bookmark Lookup		The Bookmark Lookup operator uses a bookmark (row ID or clustering key) to look up the corresponding row in the table or clustered index. Bookmark Lookup is not used in SQL Server 2014. The Key Lookup operator also provides this functionality.












Branch Repartition	None	The Branch Repartition operator is a logical operator showing where iterators could be executed by parallel threads.
Broadcast	None	The Broadcast operator sends the set of input rows to multiple consumers.
Build Hash		The Build Hash operator indicates that a batch has table has been built.
Cache	None	The Cache operator is a logical operator that caches a single row of data. Cache is not used in SQL Server 2014.
Clustered Index Delete		The Clustered Index Delete operator deletes rows from a clustered index. Clustered Index Delete is a physical operator.
Clustered Index Insert		The Clustered Index Insert operator inserts rows into a clustered index. Clustered Index Insert is a physical operator.
Clustered Index Merge		The Clustered Index Merge operator merges a data stream and a clustered index. Clustered Index Merge is a physical operator.
Clustered Index Scan		The Clustered Index Scan operator scans a clustered index. It returns only rows matching the WHERE clause and sorts the results based on the ORDERED clause.
Clustered Index Seek		The Clustered Index Seek operator retrieves rows from a clustered index using the seeking ability of indexes.
Clustered Index Update		The Clustered Index Update operator updates rows in the clustered index. It only updates rows matching the WHERE clause.
Collapse		The Collapse operator merges separate operations into a single more efficient operation.

Columnstore Index Scan		The Columnstore Index Scan operator scans the columnstore index specified in the query execution plan argument.
Compute Scalar		The Compute Scalar operator returns a computer scalar value from an evaluated expression.
Concatenation		The Concatenation operator returns rows scanned from multiple inputs. This is used for statements like UNION ALL.
Constant Scan		The Constant Scan operator adds rows into a query.
Convert		The Convert operator converts one scalar type to another.
Cross Join	None	The Cross Join operator is a logical operator that joins each row from input with each row from another input.
catchall		The catchall operator is a placeholder icon when an operator does not match any other query operators.
Cursor	None	The Cursor operator describes the execution of a query or update that use cursor operations.
Declare		The Declare operator allocates a local variable.
Delete		The Delete operator deletes rows from an object.
Deleted Scan		The Deleted Scan operator scans the deleted table within a trigger.
Distinct	None	The Distinct operator removes duplicate rows.
Distinct Sort	None	The Distinct Sort operator removes duplicate rows and sorts the resulting set of rows.

Distribute Streams		The Distribute Streams operator is used to break records from a single input to multiple output streams in a parallel query plan.
Dynamic		The Dynamic operator uses a cursor to see all changes made by others.
Eager Spool		The Eager Spool operator stores each row in the input to tempdb so that this cached data can be used if the operator is rewound.
Fetch Query		The Fetch Query operator returns rows from a fetch issued against a cursor.
Filter		The Filter operator returns only rows that match a filter expression.
Flow Distinct	None	The Flow Distinct operator removes duplicate rows but returns each row as it is processed.
Full Outer Join	None	The Full Outer Join operator is a logical operator that implements an OUTER JOIN. Results in all matching rows from two streams plus rows for each row in the streams that did not have matches.
Gather Streams		The Gather Streams operator is used to consume multiple input streams and combine them into a single output stream in parallel query plans.
Hash Match		The Has Match operator creates a hash table from the rows in the build input.
Inner Join	None	The Inned Join operator returns rows that satisfy the JOIN of the first (top) input with the second (bottom) input.
Insert		The Insert logical operator inserts input rows into the object specified in the Argument column.
Inserted Scan		The Inserted Scan operator scans the inserted table.

Intrinsic		The Intrinsic operator runs an internal Transact-SQL function.
Iterator		The Iterator operator is a placeholder icon used when no match is found for the iterator operation.
Key Lookup		The KEY LOOKUP operator signals a lookup using a bookmark on a table with a clustered index.
Keyset		The KEYSET operator uses a cursor that can only see updates and not inserts.
Language Element		The LANGUAGE ELEMENT operator is a placeholder icon when no matching language constructs can be found.
Lazy Spool		The LAZY SPOOL operator stores each row in the input to tempdb so that this cached data can be used if the operator is rewound. The rows are only copied to tempdb as they are processed.
Left Anti Semi Join	None	The LEFT ANTI SEMI JOIN operator returns rows from the first input where there is no match in the second input.
Left Outer Join	None	The LEFT OUTER JOIN operator returns rows from the first input that satisfy the join with the second input.
Left Semi Join	None	The LEFT SEMI JOIN operator returns rows from the first input that have a match with the second input.
Log Row Scan		The LOG ROW SCAN operator scans the transaction log.
Merge Interval		The MERGE INTERVAL operator merges multiple intervals to create a minimal set of non-overlapping intervals.
Merge Join		The MERGE JOIN operator performs one of the many possible join operations.

Nested Loops		The NESTED LOOPS operator perform the logical operations to satisfy many join operations that need a loop to search the inner table for rows in the outer table.
Nonclustered Index Delete		The NONCLUSTERED INDEX DELETE operator deleted rows from a non-clustered index.
Index Insert		The INDEX INSERT operator inserts rows into a non-clustered index.
Index Scan		The INDEX SCAN operator scans a non-clustered index. It returns only rows matching the WHERE clause
Index Seek		The INDEX SEEK operator retrieves rows from a non-clustered index using the seeking ability of indexes.
Index Spool		The INDEX SPOOL operator copies input rows to tempdb and builds a non-clustered index for these rows.
Nonclustered Index Update		The NONCLUSTERED INDEX UPDATE operator updates rows from its input in the non-clustered index.
Online Index Insert		The ONLINE INDEX INSERT operator is a physical operator indicating that an index create, alter or drop is performed online.
Parallelism	None	The PARALLELISM operator the logical operations of distribute, gather or repartition streams.
Parameter Table Scan		The PARAMETER TABLE SCAN operator scans a table that acts as a parameter of the query.
Partial Aggregate	None	The PARTIAL AGGREGATE operator is a logical operator that aggregates input rows to prevent writing to disk in parallel plans.
Population Query		The POPULATION QUERY operator populates a cursor's work table.

Refresh Query		The REFRESH QUERY operator fetches current data for rows.
Remote Delete		The REMOTE DELETE operator deletes rows from a remote object.
Remote Index Scan		The REMOTE INDEX SCAN operator scans a remote index.
Remote Index Seek		The REMOTE INDEX SEEK operator retrieves rows using a remote index.
Remote Insert		The REMOTE INSERT operator inserts rows into a remote object.
Remote Query		The REMOTE QUERY operator submits a query to a remote source.
Remote Scan		The REMOTE SCAN operator scans a remote object.
Remote Update		The REMOTE UPDATE operator updates a remote object.
Repartition Streams		The REPARTITION STREAMS operator creates multiple out streams from multiple input streams while applying a bitmap filter.
Result		The RESULT operator is the query plan return data.
RID Lookup		The RID LOOKUP operator is a bookmark lookup on a heap.
Right Anti Semi Join	None	The RIGHT ANTI SEMI JOIN operator returns rows from the second input where there is no match in the first input.
Right Outer Join	None	The RIGHT OUTER JOIN operator returns rows from the second input that satisfy the join with the first input.











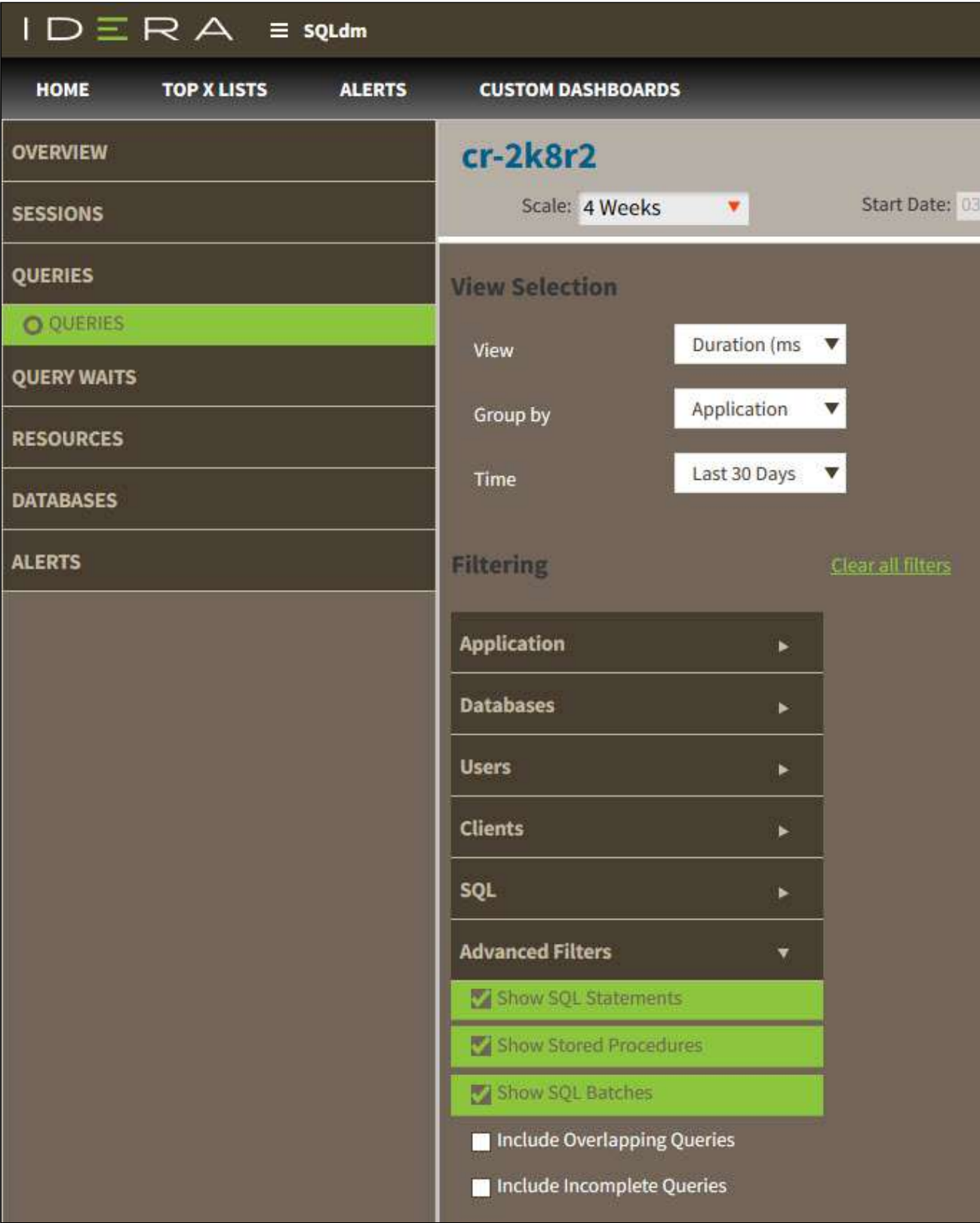
Right Semi Join	None	The RIGHT SEMI JOIN operator returns rows from the second input that have a match with the first input.
Row Count Spool		The ROW COUNT SPOOL operator returns empty rows for each row in the input stream.
Segment		The SEGMENT operator uses the value of columns to divide the input set into segments.
Segment Repartition	None	The SEGMENT REPARTITION operator marks the boundaries of regions whose iterators can be run in parallel threads.
Sequence		The SEQUENCE operator executes each input in sequence.
Sequence Project		The SEQUENCE PROJECT operator adds columns to the input set, divides the input set into segments and outputs one segment at a time.
Snapshot		The SNAPSHOT operator creates a cursor that cant see changes by others.
Sort		The SORT operator sorts incoming rows.
Split		The SPLIT operator creates a delete and insert operation out of each update operation.
Spool		The SPOOL operator saves an intermediate query to tempdb.
Stream Aggregate		The STREAM AGGREGATE operator groups rows by columns and calculates aggregate expressions.
Switch		The SWITCH operator copies the appropriate input stream to the output stream by evaluating an expression.
Table Delete		The TABLE DELETE operator deletes rows from a table.

Table Insert		The TABLE INSERT operator inserts rows into a table.
Table Merge		The TABLE MERGE operator applies a merge data stream to a heap.
Table Scan		The TABLE SCAN operator retrieves rows from a table.
Table Spool		The TABLE SPOOL operator scans the input and places the rows into tempdb.
Table Update		The TABLE UPDATE operator updates rows in a table.
Table-valued Function		The TABLE-VALUED FUNCTION operator evaluates a table-valued function and stores the resulting rows in tempdb.
Top		The TOP operator returns only the specified number of rows from the input.
Top N Sort	None	The TOP N SORT operator returns only the specified number of rows from the input and sorts them.
UDX		The UDX operator implements XQuery and XPath operations.
Union	None	The UNION operator combines multiple inputs and removes duplicates.
Update		The UPDATE operator updates a specified object from the rows in its input.
While		The WHILE operator represents a SQL while loop.
Window Spool		The WINDOW SPOOL operator expands the input rows into sets of rows that represent the window associated with the row.

Advanced filters of the Main Query view

Advanced filter options allow you to determine the precise Query Monitor data that is displayed. You can find the advanced filter options at the left side of the Main Query view. [Click image to view full size.](#)



The Advanced Filters include:

Show SQL Statements

Includes queries that are SQL Statements. This filter is enabled by default.

Show Stored Procedures

Includes queries that are Stored Procedure calls. This filter is enabled by default.

Show SQL Batches

Includes queries that are SQL batches. This filter is enabled by default.

Include Overlapping Queries

Includes queries whose total period of execution overlaps with the selected time period. This filter is disabled by default.

Include Incomplete Queries

Includes queries whose total period of execution exceeds the selected time period (i.e. query information is incomplete). This filter is disabled by default.

List view of the main query view

The list view displays all matching objects for the [view selection](#) and [grouping](#) in the Main Query view. The following list options are available:

- [Application list](#)
- [Database list](#)
- [User list](#)
- [Client list](#)
- [Query Signature list](#)

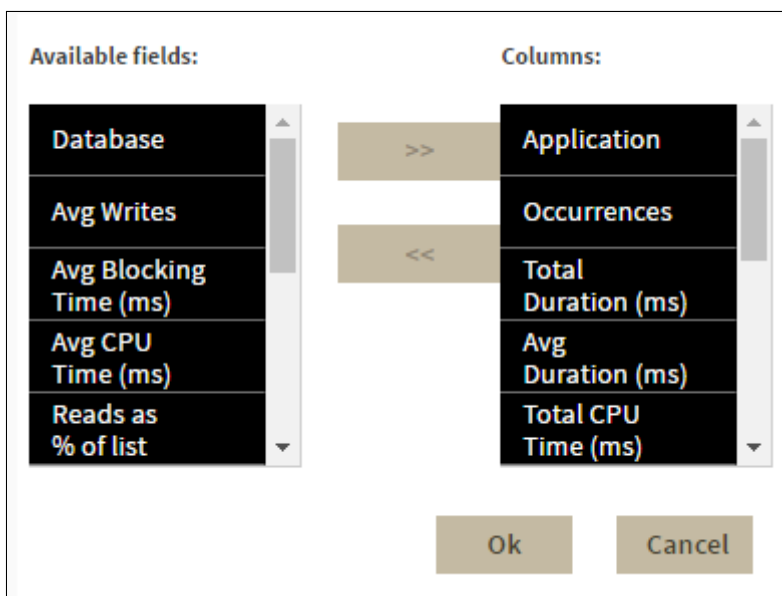
The screenshot shows the IDERA SQLdm interface. The top navigation bar includes 'HOME', 'TOP X LISTS', 'ALERTS', and 'CUSTOM DASHBOARDS'. The main content area is titled 'cr-2k8r2' and features a 'Scale: 4 Weeks' dropdown and a 'Start Date' field. The left sidebar contains a menu with 'OVERVIEW', 'SESSIONS', 'QUERIES' (highlighted), 'QUERY WAITS', 'RESOURCES', 'DATABASES', and 'ALERTS'. The 'QUERIES' section is active, displaying a 'View Selection' dropdown menu with options: 'Duration (ms)', 'Application' (selected), 'Database', 'User', 'Client', and 'Query Signature'. Below this is a 'Filtering' section with expandable categories: 'Application', 'Databases', 'Users', 'Clients', 'SQL', and 'Advanced Filters'. The 'Advanced Filters' section includes checkboxes for 'Show SQL Statements', 'Show Stored Procedures', 'Show SQL Batches', 'Include Overlapping Queries', and 'Include Incomplete Queries'. A 'Clear all filters' link is visible to the right of the filtering options.

Application list

The application list displays data that results from grouping by Application in the View Selection of the Main Query view. The application list provides you with the following information by default:

- Application
- Occurrences
- Total Duration (ms)
- Average Duration (ms)
- Total CPU Time (ms)
- Total Reads
- Total Writes
- Total I/O
- Total Wait Time (ms)
- Most Recent Completion
- Total Blocking Time (ms)
- Total Deadlocks

You can add or remove columns through the Column Chooser under **Options**.



- Database
- Average Writes
- Average Blocking Time (ms)
- Average CPU Time (ms)
- Reads as % of list
- Average I/O
- CPU as % of list
- Average Reads
- Average CPU Per Second
- Average Deadlocks
- Average Wait Time (ms)

✓ Records

You can display the top 100, 200, or all records by clicking the different **Show Top** options.

Explanation: CPU as % of List

In the Main Query view, after selecting the CPU Time (ms) by Application view. The CPU as % of list metric provides the CPU as a percentage of the values returned in the application list. For example, if you have three applications returned with a total CPU Time (ms) of 600, then the CPU as % of list metric for each is calculated as follows:

Application 1- CPU Time (ms): 100; $(100/600) * 100 = 16.7\%$ of total application CPU.

Application 2- CPU Time (ms): 200; $(200/600) * 100 = 33\%$ of total application CPU.

Application 3- CPU Time (ms): 300; $(300/600) * 100 = 50\%$ of total application CPU.

The CPU as % of list metric helps to pinpoint which application in your list has the highest CPU Time percentage (Application 3).

To drill down for additional query information, you can click the gear icon in the list and the following context menu displays:

- **Query Signature.** Click this option to view query signatures for the selected application.
- **Query Statement.** Click this option to view query statements for the selected application.

i To access the Query Signature view, you can also click the linkable application name (e.g. Microsoft SQL Server) in the application list.

Database list

The database list displays data that results from grouping by Database in the View Selection of the Main Query view. The database list provides you with the following information by default:

- Database
- Occurrences
- Total Duration (ms)
- Average Duration (ms)
- Total CPU Time (ms)
- Total Reads
- Total Writes
- Total I/O
- Average Wait Time (ms)
- Most Recent Completion
- Average Blocking Time (ms)
- Average Deadlocks

Additional columns are available through the Column Chooser under **Options**.

- Average Writes
- Total Blocking Time (ms)
- Average CPU Time (ms)
- Reads as % of list
- Total Wait Time (ms)
- Average I/O
- Total Deadlocks
- CPU as % of list
- Average Reads

- Average CPU Per Second
- Application

To drill down for additional query information, you can click the gear icon in the list and the following context menu displays:

- **Query Signature.** Click this option to view query signatures for the selected database.
- **Query Statement.** Click this option to view query statements for the selected database.



Records

You can display the top 100, 200, or all records by clicking the different **Show Top** options.



To access the Query Signature view, you can also click the linkable database name in the database list.

User list

The user list displays data that results from grouping by the user in the View Selection of the Main Query view. The user list provides you with the following information by default:

- Users
- Occurrences
- Total Duration (ms)
- Total CPU Time (ms)
- Total Reads
- Total Writes
- Average Wait Time (ms)
- Most Recent Completion
- Average Blocking Time (ms)
- Average Deadlocks

Additional columns are available through the Column Chooser under **Options**:


- Database
- Average Writes
- Total Blocking Time (ms)
- Average Duration (ms)
- Average CPU Time (ms)
- Reads as % of list
- Total Wait Time (ms)
- Average I/O Per Second
- Total Deadlocks
- CPU as % of list
- Average Reads
- Average CPU Per Second
- Application

To drill down for additional query information, you can click the gear icon in the list and the following context menu displays:

- **Query Signature.** Click this option to view query signatures for the selected user.
- **Query Statement.** Click this option to view query statements for the selected user.

 **Records**

You can display the top 100, 200 or all records by clicking the different **Show Top** options.

 To access the Query Signature view, you can also click the linkable username in the user list.

Client list

The client list is the data displayed after you selected **Group By** Clients in the View Selection of the Main Query view. The client list provides you with the following information by default:

- Clients
- Occurrences
- Total Duration (ms)
- Average Duration (ms)
- Total CPU Time (ms)
- Total Reads
- Total Writes
- Average Wait Time (ms)
- Most Recent Completion
- Average Blocking time (ms)
- Average Deadlocks

Additional columns are available through the Column Chooser under **Options**.


- Database
- Average Writes
- Total Blocking Time (ms)
- Average CPU Time (ms)
- Reads as % of list
- Total Wait Time (ms)
- Average I/O Per Second
- Total Deadlocks
- CPU as % of list
- Average Reads
- Average CPU Per Second
- Application

To drill down for additional query information, you can click the gear icon in the list and the following context menu displays:

- **Query Signature.** Click this option to view query signatures for the selected client.
- **Query Statement.** Click this option to view query statements for the selected client.

 **Records**

You can display the top 100, 200 or all records by clicking the different **Show Top** options.

 To access the Query Signature view, you can also click the linkable client name in the client list.

Query Signature list

The Query signature list is the data displayed after you selected **Group By** Query Signature in the View Selection of the Main Query view. The Query signature list provides you with the following information by default:

- Query ID
- Signature SQL Text
- Occurrences
- Event Type
- Total Duration (ms)
- Total CPU Time (ms)
- Total Reads
- Total Writes
- Average Wait Time (ms)
- Most Recent Completion
- Average Blocking Time (ms)
- Average Deadlocks
- Keep Detailed History Flag
- Aggregated

Additional columns are available through the Column Chooser under **Options**.

- Database
- Average Writes
- Total Blocking Time (ms)
- Average Duration (ms)
- Average CPU Time (ms)
- Reads as % of list
- Total Wait Time (ms)
- Average I/O Per Second
- Total Deadlocks
- CPU as % of list
- Average Reads
- Average CPU Per Second
- Application

To research for additional query details, you can perform the following actions from this list:

- [View SQL](#)
- [Query Statement](#)

 You can display the top 100, 200, or all records by clicking the different **Show Top** options.

View your SQL Server query waits information

The Query waits tab of the single instance dashboard allows you to analyze waits over time (**Query Waits over Time**) and by duration (**Query Waits by Duration**), so that you can better determine where your biggest bottlenecks are occurring and what changes could potentially have the biggest performance boost on your SQL Server instance.

Query waits over time

The Query waits over time view provides a selectable time range for the stacked bar chart display of waits. This view comes with various dimensions for display that allow you to further investigate any performance issues:

- Waits
- Waits Category
- Statements
- Applications
- Databases
- Clients
- Sessions
- Users

Access the Query waits by duration view

You can navigate to the Query waits by duration view by double-clicking any bar in this view.

Query waits by duration

The Query waits by duration view displays a horizontal bar chart of query waits by duration of the wait. This view comes with various dimensions for display that allow you to further investigate any performance issues:

- Waits
- Waits Category
- Statements
- Applications
- Databases
- Clients
- Sessions
- Users

Time range

In the Query waits by duration view, the horizontal bar chart is filtered by the selected time range in the Query waits over time view.

Access the query waits tab

To access the query waits enhanced views, click the monitored instance of your choice in the Home tab of the SQL Diagnostic Manager web console, and then select the Query Waits tab.

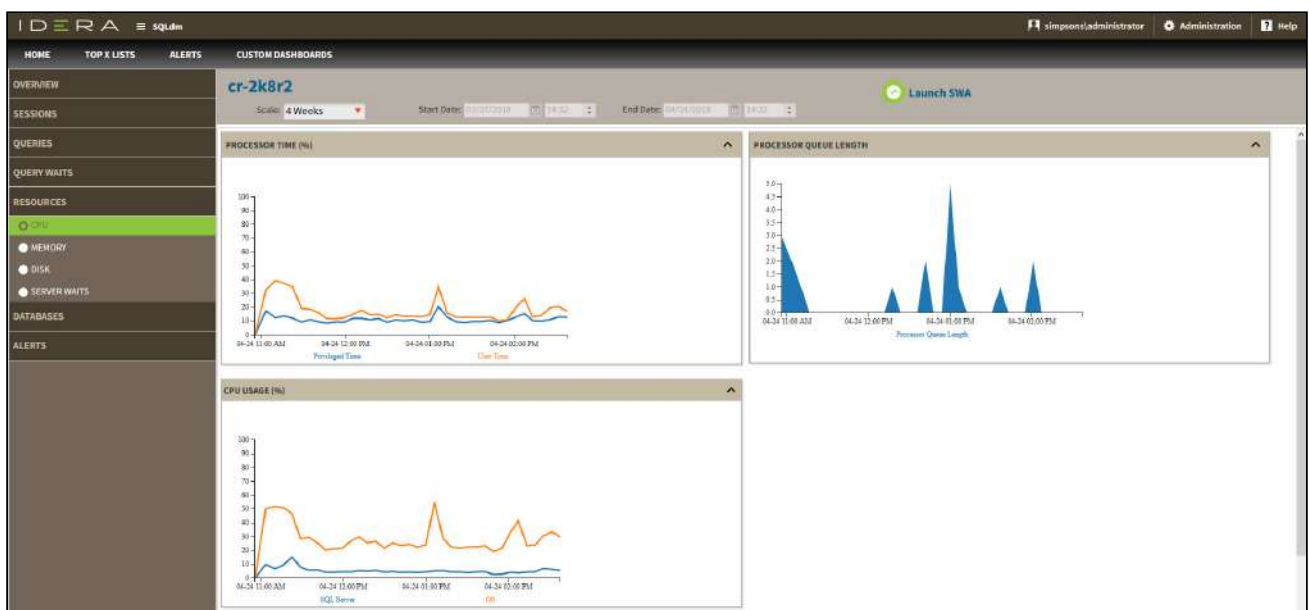
View your SQL Server resources information

The Resources tab of the single instance dashboard contains several views such as [CPU](#), [Memory](#), [Disk](#), and [Server Waits](#). The Resources tab allows you to monitor the system resources on the computer hosting the SQL Server instance as well as what is used by SQL Server.

Resource bottlenecks are often the cause of SQL Server performance problems. These bottlenecks can result from poor database application design or point to other system processes using valuable resources needed by SQL Server.

CPU view

The **Resources > CPU** tab contains several charts and a Top sessions by active CPU widget (Click image to view full size).



Charts

Processor Time (%)

The Processor Time chart displays the percentage of CPU time used by all processes on the computer hosting the SQL Server instance.


Processor Queue Length

The Processor Queue Length chart displays the number of ready threads in the processor queue on the computer hosting the SQL Server instance.

CPU Usage (%)

The CPU Usage (%) chart displays the percentage of processing power in use on the computer that hosts the SQL Server instance over a period of time. You can select that information displays for the last 30 minutes, 1 hour, 4 hours, 8 hours, and 12 hours.

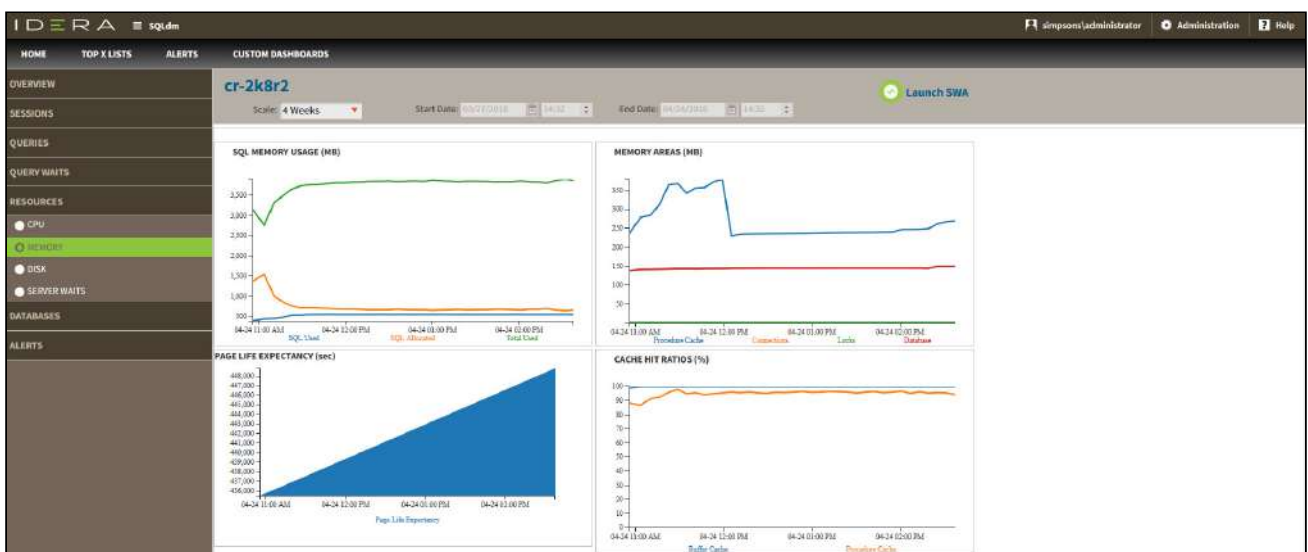
Top Sessions by Active CPU

This widget allows you to view the top sessions of a selected SQL Server instance per CPU usage. You can configure this widget to display additional entries by clicking the gear icon or you can expand/contract by clicking .

Session ID	Host	Database	CPU Usage
56	cr-2k8r2	master	93,956.00
57	cr-2k8r2	master	13,131.00
50	cr-2k8r2	master	8,124.00
54	cr-2k8r2	master	873.00
53	cr-2k8r2	master	484.00

Memory view

The **Resources > Memory** tab contains the following charts (Click image to view full size).



Charts

SQL Memory Usage (MB)

This chart displays the SQL memory usage (SQL Used) compared to what is allocated (SQL Allocated) and the total used (Total Used).

Memory Areas (MB)

This chart displays the memory usage of the Procedure Cache, Connections, Locks, and Database.

Page Life Expectancy (sec)

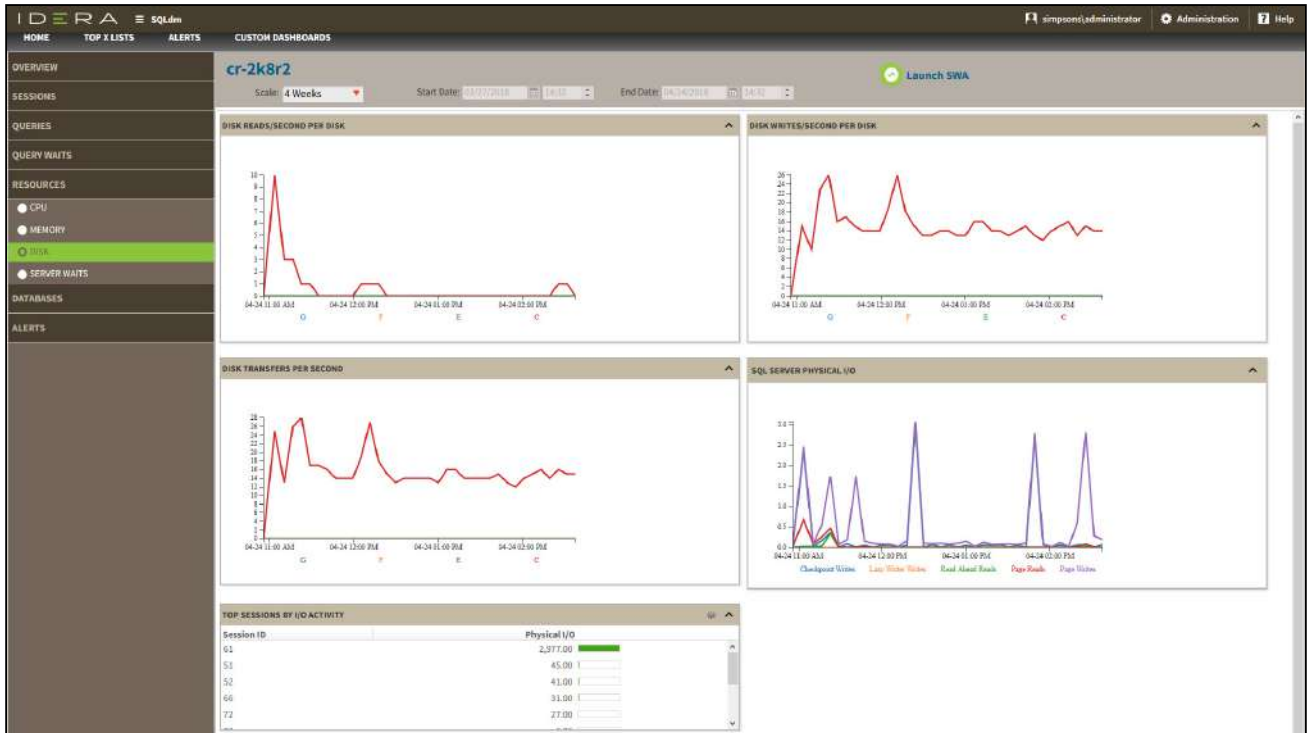
This chart provides the time in seconds that a page remains in the cache or buffer pool.

Cache Hit Ratios (%)

This chart displays the hit ratios of the Buffer and Procedure cache.

Disk view

The **Resources > Disk** tab contains several charts and the Top sessions by I/O activity widget (Click image to view full size).



Charts

Disk Reads/Second Per Disk

This chart displays the number of reads per disk per second during a sample interval. You can select intervals of 30 minutes, 1 hour, 4 hours, 8 hours, and 12 hours.

Disk Writes/Second Per Disk

This chart displays the number of writes per disk per second during a sample interval. You can select intervals of 30 minutes, 1 hour, 4 hours, 8 hours, and 12 hours.


Disk Transfers Per Second






This chart displays the number of disk transfers per second during a sample interval. You can select intervals of 30 minutes, 1 hour, 4 hours, 8 hours, and 12 hours.

SQL Server Physical I/O

This chart displays the number of read and write requests sent to the physical disk the SQL Server instance resides in. Read and write requests are classified in Checkpoint Writers, Read-Ahead Reads, Page Writes, Lazy Writer Writes, and Page Reads.

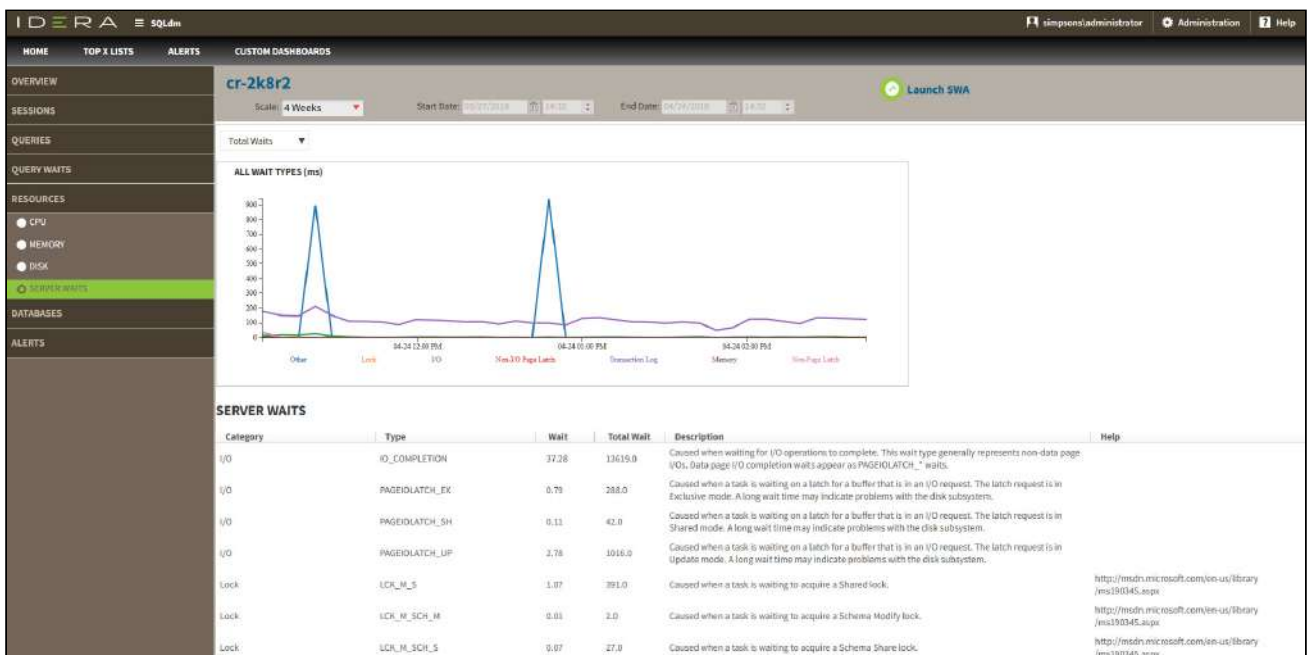
Top Sessions by I/O Activity

This widget allows you to view the top sessions of a selected SQL Server instance per I/O Activity. You can configure this widget to display additional entries by clicking the gear icon or you can expand/contract by clicking .

Session ID	Physical I/O
57	1,637.00 
51	273.00 
60	101.00 
53	62.00 
58	10.00 

Server Waits view

The **Resources > Server Waits** tab contains a chart that display information on all waits occurring on your system and show the time different wait types occurred and the length of time (in milliseconds) it took for the waits to clear. Click image to view full size.



Wait categories in the drop-down menu

Total Waits

Total waits include all the types of waits collected.

Signal Waits

Signal waits occur when a resource is now available (after a resource wait) and the task is waiting to run again.

Resource Waits

Resource waits occur when access to a resource is denied because the resource is in use or not available.

Server Waits panel

Below the charts, the Server Waits panel includes information on the wait category, wait type, length of time the wait took to clear, total wait time for all the collected wait types, and a definition of the wait type.

Available wait types

The SQL Diagnostic Manager web console displays information on the following wait types: I/O, Memory, Non-I/O Page Latch, Non-Page Latch, Transaction Log, and Other categories.

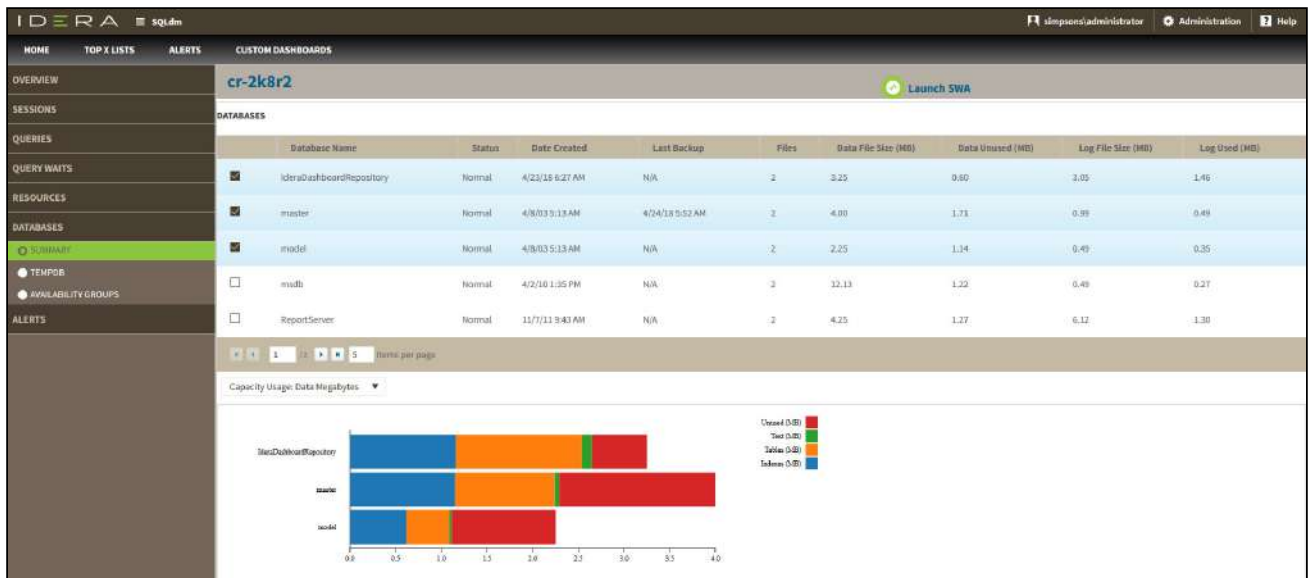
For additional information on Server Waits metrics, see [Server Waits Panel](#).

Access the Resources tab

SQL Diagnostic Manager provides two paths to access the Resources tab. The first access path is by selecting the Resources tab of the single instance dashboard. You can also click the resources category icon of a specific instance in the thumbnail sub-view.

View your SQL Server databases information

The Databases tab of the single instance dashboard contains the [Summary](#), [Tempdb](#), and [Availability Groups](#) views. The Databases tab allows you to view the status of all the databases on the selected SQL Server instance, the tempdb database, and the availability groups.



Summary view

The **Databases > Summary** tab lists each database on the monitored SQL Server instance and provides a large variety of database statistics:

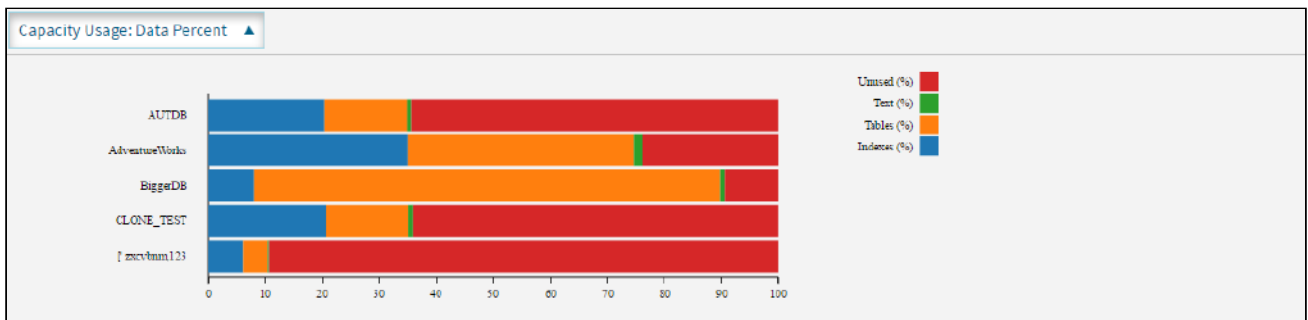
- Database name
- Status
- Date Created
- Last Backup
- Files
- Data File Size (MB)
- Data Unused (MB)
- Log File Size (MB)
- Log Used (MB)

✓ Multi-database selection

Select multiple databases to display as horizontal stacked bar charts.

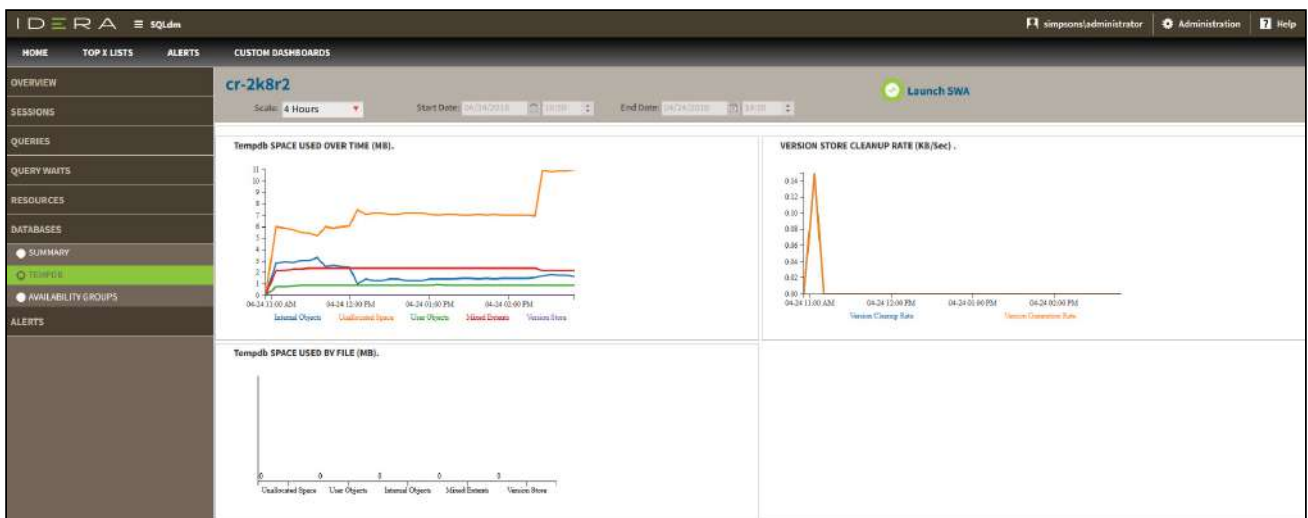
Capacity Usage chart

You can choose to view how data or log is used on your database. The Capacity Usage drop-down list allows you to choose data or log in megabytes or percentages. You can select one or more databases in the Databases panel to display in the Capacity Usage chart. This chart provides information on the text, tables, indexes space, as well as the unused space of your database.



Tempdb view

The **Databases > Tempdb** tab allows you to view the status of your tempdb database on the selected SQL Server instance. Click image to view full size.



The Tempdb view contains the following charts:

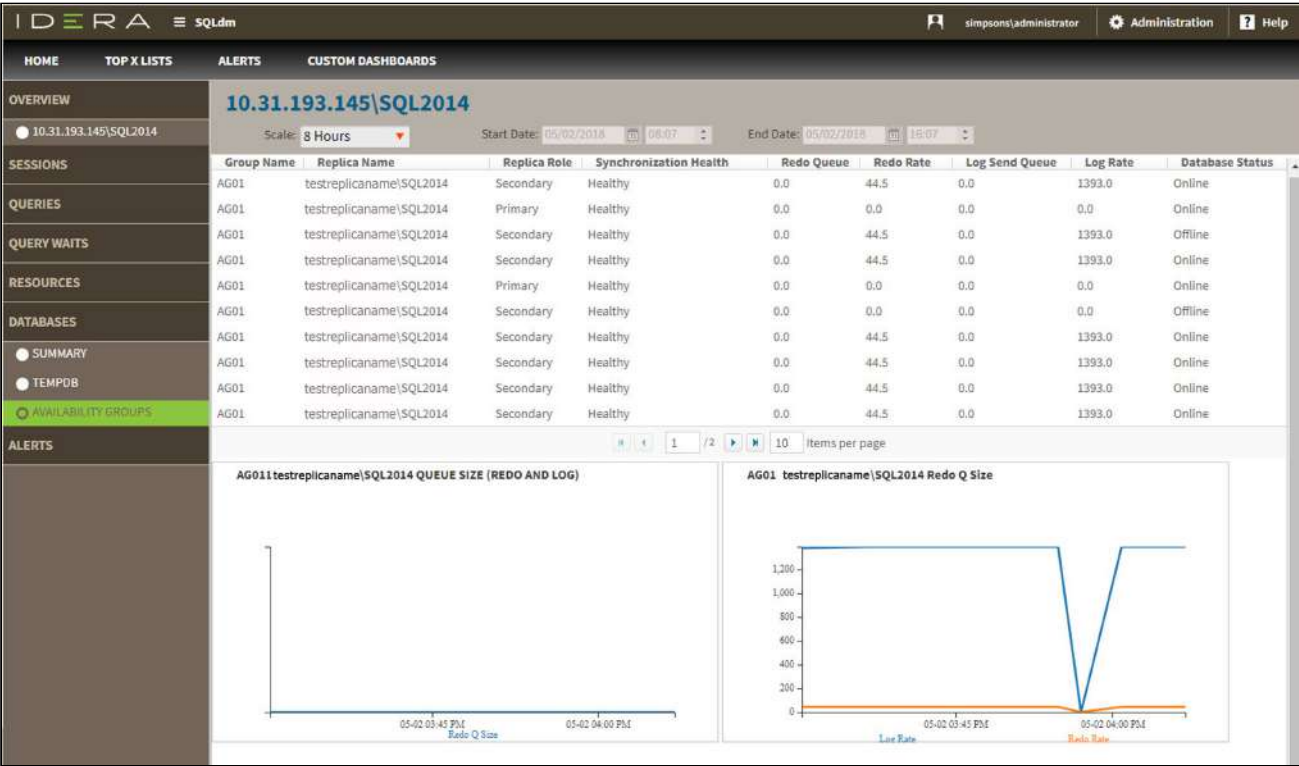
- Tempdb Space Used over Time (MB)
- Version Store Cleanup Rate (KB/Sec)
- Tempdb Space Used by File (MB)

For information on each chart, see [Get the tempdb status summary](#).

Availability Groups

The **Databases > Availability Groups** tab allows you to view the status of your availability groups on the selected SQL Server instance. This tab lists the Availability Groups, with the following description:

- Group Name
- Replica Name
- Replica Role
- Synchronization Health
- Redo Queue
- Redo Rate
- Log Send Queue
- Log Rate
- Database Status



Access the Databases tab

The SQL Diagnostic Manager web console provides two paths to access the Databases tab. The first access path is by selecting the Databases tab of the single instance dashboard. You can also click the database category icon of a specific instance in the thumbnail view.

View your SQL Server alerts

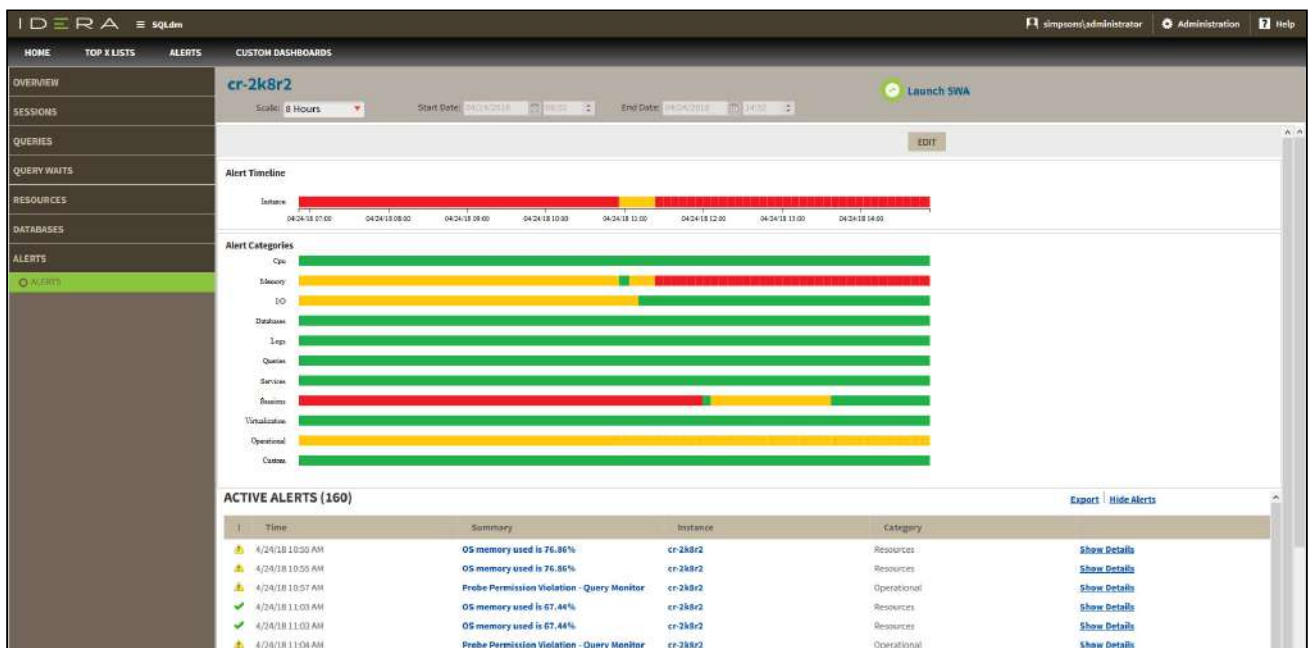
The Alerts tab of the single instance dashboard displays a color-coded alerts timeline view and an active alerts panel for the selected instance.

Color-coded alerts timeline view

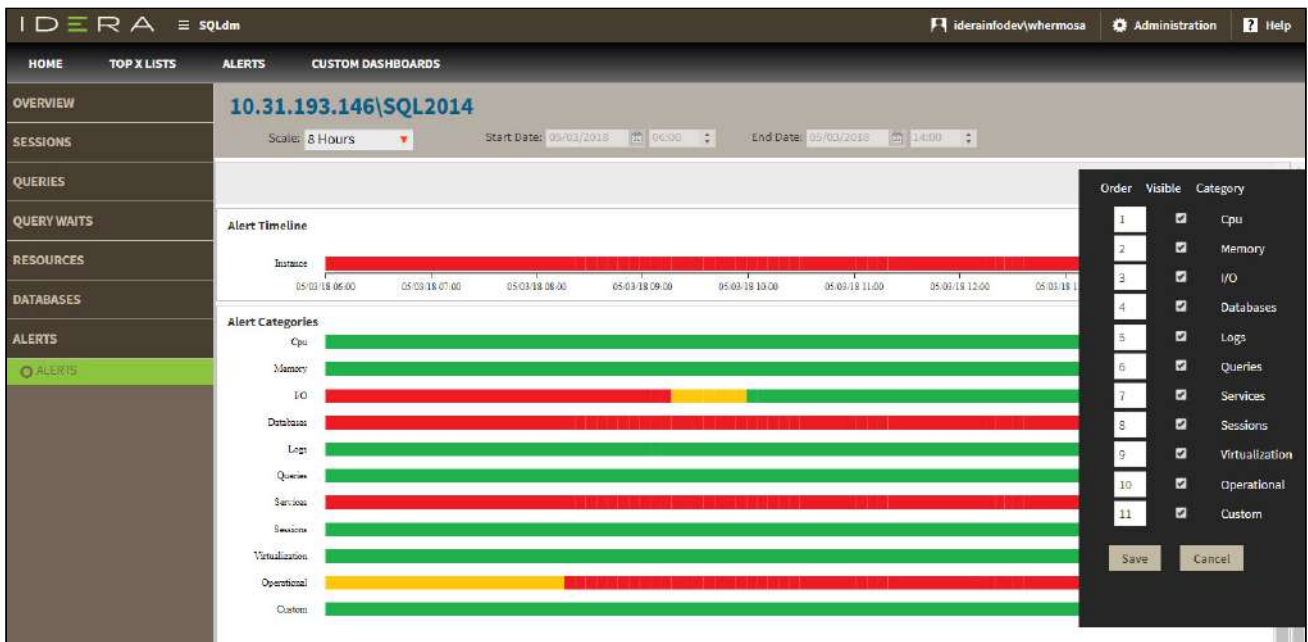
The color-coded alerts timeline view allows you to easily identify the highest alert for your monitored SQL Server instance and the highest alert for a specific category. Each color in the alert timeline represents a different alert type. See the following table for more details:

Alert type	Color	Meaning
OK	Green	The alert is in an acceptable threshold.
Warning	Yellow	The alert reaches a Warning threshold.
Critical	Red	The alert reaches a Critical threshold.

Filter your alert results in the Alerts tab by specifying a timeframe or by selecting default timeframes of 8 hours, 1 day, 5 days, or 4 weeks. Click image to display in full size view.



You can also filter your alert results by alert category or categories' order. Click **Edit** to order and select the categories you want to display in the Alerts timeline view.



Active alerts panel

The Alerts tab also lists all the active alerts of your monitored SQL Server instance over a period of time. The active alerts panel provides the following information:

Alert type

Shows alert type icon:  for OK alerts,  for Warning alerts, and  for Critical alerts.

Time

Indicates date and time of alert inception.

Summary

Displays alert description.

Instance

Displays the name of the monitored SQL Server instance.

Category

Indicates alert category.

Additional options

In the Active alerts panel of the single instance dashboard, you can select the following additional options:

Show alert details

View [additional information](#) on a specific active alert by clicking **Show Details**.

Export

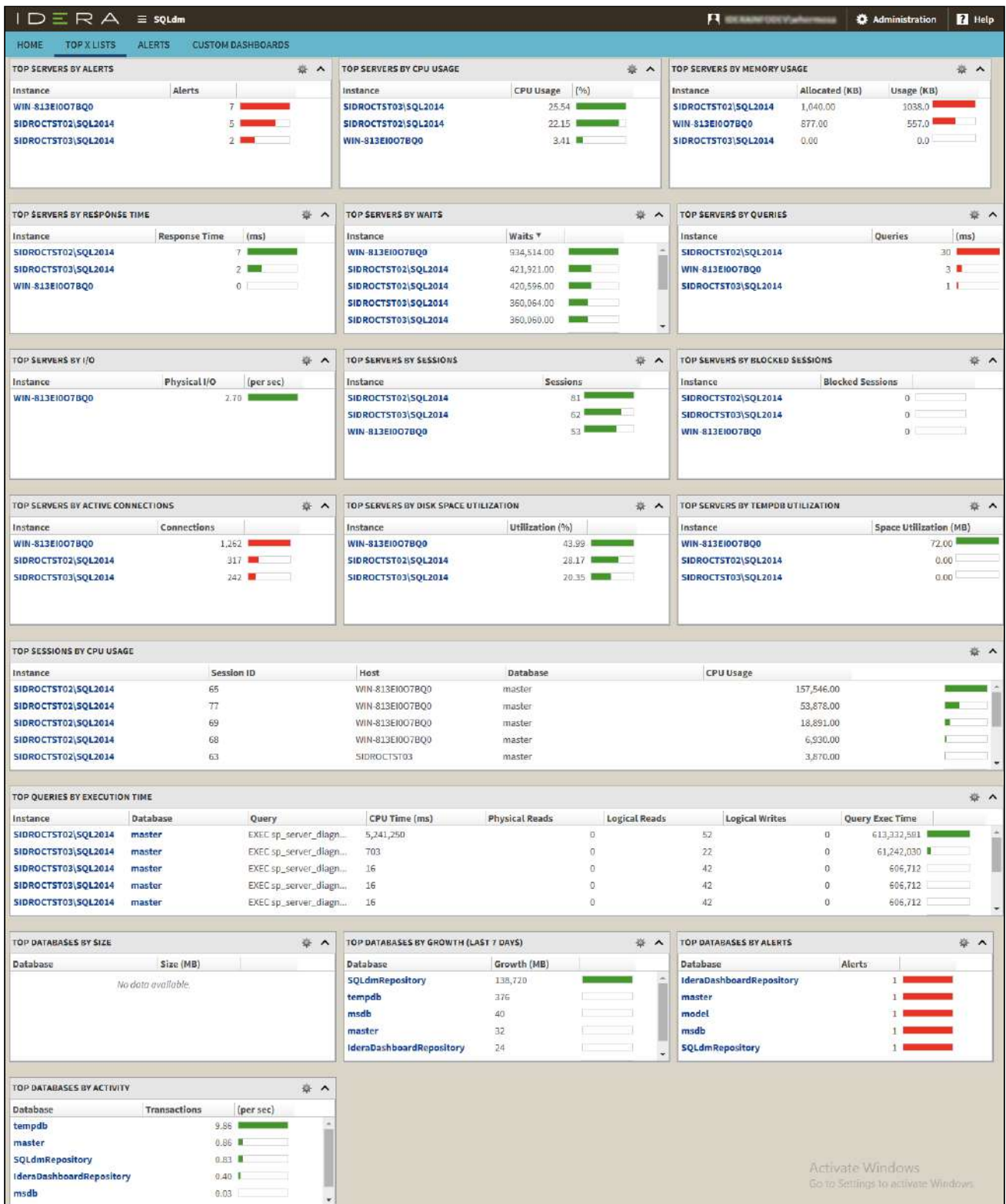
Export your alerts information in PDF, XLS, and XML.

Show/Hide Alerts

Choose to show or hide active alerts.

View your top values in the web console

The Top X Lists tab of the SQL Diagnostic Manager web console allows you to view the top values in descending order for selected metrics across the instances and databases in your SQL Server environment. Click the image to view full size.



The **Top X Lists dashboard** contains 18 widgets that help you assess quickly the health and performance of your SQL Server environment. You can expand widgets, drill down for details, and also configure them as follows:

Records

Choose to display a particular number of records with a maximum of 50. Access this option through the gear icon on the right of the widget.

On the **Top X Lists dashboard** you can find the following widgets:

Top servers by alerts

This widget allows you to view the instances with the highest number of alerts during the past seven days in your SQL Server environment. This widget displays the following information:

- Instance name
- Alerts

Clicking a specific server (instance name) in the widget allows you to navigate to the **Overview** tab of the single instance dashboard for that server.

Top servers by CPU usage

This widget allows you to view the instances with the highest percentage of CPU usage in your SQL Server environment. This widget displays the following information:

- Instance name
- CPU usage in percentage

Clicking a specific instance name in the widget allows you to navigate to the **Resources > CPU** tab of the single instance dashboard for that server.

Top servers by memory usage

This widget provides you with information on the top instances with the highest percentage of memory usage in comparison to their SQL Server allocated memory. This widget displays the following information:

- Instance name
- Allocated (KB)
- Usage (KB)

Clicking a specific server (instance name) in the widget allows you to navigate to the **Resources > Memory** tab of the single instance dashboard for that server.

Top Servers by response time

This widget provides you with information on the top response times in milliseconds for all instances in your environment. Keeping a tab of the top response times helps monitor congestion in a network and the performance of SQL Server.

This widget displays the following information:

- Instance name
- Response time in milliseconds

Clicking a specific server (instance name) in this widget allows you to navigate to the overview tab of the single instance dashboard for that server.

Top Servers by waits

This widget provides you with information on the top waits for all instances in their environment. This widget displays the following information:

- Instance name
- Waits

Clicking a specific server (instance name) in the widget allows you to navigate to the **Resources > Server Waits** tab of the single instance dashboard for that server.

Top servers by queries

This widget allows you to view the top instances with the highest number of queries in your SQL Server environment. Keeping a tab of this metric in your environment lets you prevent overloading your server.

This widget displays the following information:

- Instance name
- Queries (ms)

Clicking a specific server (instance name) in the widget allows you to navigate to the main dashboard.

Top servers by I/O

This widget provides you with information on the top 10 instances with the highest I/O load by SQL Physical I/O (reads and writes) in their SQL Server environment. Keeping tab of the I/O subsystem performance prevents increased timeouts and response times.

This widget displays the following information:

- Instance name
- Physical I/O (per sec)

Clicking on a specific server (instance name) in the widget allows you to navigate to the **Resources > Disk** tab of the single instance dashboard for that server.

Top servers by sessions

This widget allows you to view the top values of the total number of sessions by instance in your SQL Server environment. Keeping a tab of this metric in your environment lets you monitor the system load.

This widget displays the following information:

- Instance name
- Sessions

Clicking a specific server (instance name) in the widget allows you to navigate to the **Sessions > Summary** tab of the single instance dashboard for that server.

Top servers by blocked sessions

This widget allows you to view the top blocked sessions by instances in your SQL Server environment. This widget displays the following information:

- Instance name
- Blocked sessions

Clicking a specific server (instance name) in the widget allows you to navigate to the **Sessions > Overview** tab of the single instance dashboard for that server.

Top Servers by active connections

This widget provides you with information on the top number of most user connections by instance in a SQL Server environment. This widget helps you keep on top of the load on the system and monitor bottlenecks.

This widget displays the following information:

- Instance name
- Connections

Clicking a specific server (instance name) in the widget allows you to navigate to the **Sessions > Summary** tab of the single instance dashboard for that server.

Top servers by disk space utilization

This widget allows you to view the top instances with the highest percentage of disk space utilization in your SQL Server environment. Keeping a tab of this metric in your environment, lets you monitor available and used disk space so that there is enough space to be allocated to databases, databases backups, and any other types of files that you may want to store on the server.

This widget displays the following information:

- Instance name
- Utilization (%)

Clicking a specific database name in the widget allows you to navigate to the **Resources > Disk** tab of the single instance dashboard for that server.

Top servers by tempdb utilization

This widget allows you to view the top values of the instances with the most tempdb space utilization in your SQL Server environment. High tempdb utilization may lead to costly performance issues. This widget displays the following information:

- Instance name
- Space utilization (MB)

Clicking a specific server (instance name) in the widget allows you to navigate to the **Databases > Tempdb** tab of the single instance dashboard for that server.

Top sessions by CPU usage

This widget allows you to view the top values of sessions by CPU Usage in your SQL Server environment. This widget helps you monitor the system load on your SQL Server environment. This widget displays the following information:

- Instance name
- Session ID
- Host
- Status
- Database
- Session CPU Usage

Clicking a specific server (instance name) in the widget allows you to navigate to the **Sessions > Summary** tab of the single instance dashboard for that server.

Top queries by execution time

This widget allows you to view the top longest running queries in your SQL Server environment. Long running queries not only consume system resources that make the server and application run slowly, but they may also lead to table locking and data corruption.

This widget displays the following information:

- Instance name
- Database name
- Query name
- CPU time (ms)
- Physical Reads
- Logical Reads
- Logical Writes
- Query execution time (ms)

Clicking on any row in this widget allows you to navigate to the **Queries > Summary** tab of the single instance dashboard for that server.

- ✔ You can configure the Top queries by execution time widget by time frame and records to display. Click the gear icon to select a specific number of records, and choose to display the longest running queries for the **Last 24 hours** or **Last 7 days**. Note that the maximum amount of records the widget displays is 50.

Top databases by size

This widget provides you with information on the top databases by file size across your SQL Server environment. Note that values represent the actual amount of data used inside a database. This widget displays the following information:

- Database name
- Size (MB)

Clicking a specific database (database name) in the widget allows you to navigate to the main dashboard.

Top databases by growth (Last 7 days)

This widget allows you to view the top fastest growing databases by instance in your SQL Server environment. Keeping a tab of this metric in your environment, lets you predict how large databases may grow. This widget displays the following information:

- Database name
- Growth (MB)

Clicking a specific database name in the widget allows you to navigate to the main dashboard.

Top databases by alerts

This widget allows you to view the top instances with the highest number of database alerts in your SQL Server environment. This widget displays the following information:

- Database name
- Alerts

Clicking a specific database name in the widget allows you to navigate to the **Databases > Summary** tab of the single instance dashboard for that server's database.

Top databases by activity

This widget provides you with information on the top databases by activity size across your SQL Server environment. This widget displays the following information:

- Database name
- Transactions (per sec)

Clicking a specific database name in the widget allows you to navigate to the **Databases > Summary** tab of the single instance dashboard for that database.

View all your alerts in the web console

SQL Diagnostic Manager includes an improved Dashboard. The Alerts tab of the SQL Diagnostic Manager web console allows you to view all alerts for all the servers in your environment.

The screenshot shows the Alerts tab in the SQL Diagnostic Manager web console. The interface includes navigation tabs (HOME, TOP X LISTS, ALERTS, CUSTOM DASHBOARDS) and a filter bar with options like Active, Severity, Instance, Metric, Category, SQLdmRepository, and Advanced Filter. Below the filter bar is a table of alerts with columns for Time, Summary, Instance, Database, Category, and Show Details. The table lists several alerts related to database backup status for various instances.

Time	Summary	Instance	Database	Category	Show Details
4/16/2019 1:11 PM	Database IderaDashboardRepository is not backed up for 9999999 days.	WIN-U059WU9TQLQ	N/A	Backup	Show Details
4/16/2019 1:11 PM	Database ReportServer is not backed up for 9999999 days.	WIN-U059WU9TQLQ	N/A	Backup	Show Details
4/16/2019 1:11 PM	Database model is not backed up for 9999999 days.	WIN-U059WU9TQLQ	N/A	Backup	Show Details
4/16/2019 1:11 PM	Database msdb is not backed up for 9999999 days.	WIN-U059WU9TQLQ	N/A	Backup	Show Details
4/16/2019 1:11 PM	Database master is not backed up for 9999999 days.	WIN-U059WU9TQLQ	N/A	Backup	Show Details
4/16/2019 1:11 PM	Database ReportServerTempDB is not backed up for 9999999 days.	WIN-U059WU9TQLQ	N/A	Backup	Show Details
4/16/2019 1:11 PM	Database SQLdmRepository is not backed up for 9999999 days.	WIN-U059WU9TQLQ	N/A	Backup	Show Details
4/16/2019 1:13 PM	Database SQLInventoryManagerRepository is not backed up for 152 days.	SIDROCTST03\SQL2014	N/A	Backup	Show Details
4/16/2019 1:13 PM	Database ReportServer\SQL2014 is not backed up for 9999999 days.	SIDROCTST03\SQL2014	N/A	Backup	Show Details

Organize your alerts


Your alerts are organized under the following columns:

- **Active** - Organizes all your alerts that are currently active.
- **Severity** - Organizes all your alerts by their severity (Severity 8, Severity 4, and Severity 1).
- **Instance** - Organizes all your alerts by monitored SQL Server instance.
- **Metric** - Organizes all your alerts by each of the metrics affected.
- **Category** - Organizes all your alerts by their category.
- **SQLdmRepository** - Organizes all your alerts by SQL Diagnostic Manager Repository.
- [Advanced Filter](#)

Severity

In the new Alert grid, **Severity** uses the following numerical values:

- Severity 8 = Critical
- Severity 4 = Warning
- Severity 1 = Informational

The Dashboard console has a new alert grid and custom filters that help you organize your alerts in your most convenient way. Click the  icon located next to the column name to expand the drop down menu. This menu allows you to sort your criteria by Sort Ascending or Sort Descending, you can remove or add the Database, Category, and Show Details columns, and group the information by fields.


The screenshot shows the IDERA SQLdm Alerts page. The navigation bar includes HOME, TOP X LISTS, ALERTS (selected), and CUSTOM DASHBOARD. The Alerts page has a filter bar with 'Active' selected, and columns for Severity, Instance, Metric, Category, SQLdmRepository, and Advanced Filter. A download icon and 'Clear Filters' button are also present. The main table has columns for Time, Summary, and an action column. A context menu is open over the 'Time' column header, showing options: Sort Ascending, Sort Descending, Columns (with a sub-menu for Database, Category, and Show Details), Group by this field, and Show in groups. The table contains several alerts, all with a severity of 'Error' (indicated by a red 'x' icon) and a summary indicating that a database is not backed up for a certain number of days.

!	Time ↓	Summary
✖	4/16/2019 1:11 PM	boardRepository is not backed up for 9999999 days.
✖	4/16/2019 1:11 PM	er is not backed up for 9999999 days.
✖	4/16/2019 1:11 PM	99 days.
✖	4/16/2019 1:11 PM	99 days.
✖	4/16/2019 1:11 PM	999 days.
✖	4/16/2019 1:11 PM	Database ReportServerTempDB is not backed up for 9999999 days.
✖	4/16/2019 1:11 PM	Database SQLdmRepository is not backed up for 9999999 days.
✖	4/16/2019 1:13 PM	Database SQLInventoryManagerRepository is not backed up for 152 days.

Additional options

- Access the [single instance dashboard](#) by clicking the instance name in an alert row.
- Access the [Alerts Detail](#) screen by clicking **Show Details**.
- Export your alerts information in different formats: Excel, CSV, and HTML.

Alerts

Active	Severity	Instance	Metric	Category	SQLdmRepository	Advanced Filter
 ▼	Clear Filters					
Excel				Summary		
CSV				11 PM	Database IderaDashboardRepository is not backed up for 9999999 days.	
HTML				11 PM	Database ReportServer is not backed up for 9999999 days.	

Access alert details

The Alerts Detail screen in the SQL Diagnostic Manager web console provides you with detailed statistics on your alerts.

The screenshot displays the Alerts Detail screen for a warning alert. At the top left, a yellow warning icon is followed by the text "WARNING" and "Disk C is 88.12% full". To the right are navigation buttons: "<< PREV" and "NEXT >>". Below this, a table-like layout shows alert details: Instance Name (CH-SP2010-1), Value (88.1197052001953), Current alert time (9/6/16 2:35 PM), Warning Threshold (75), and Critical Threshold (90). A "Metric History" section shows a line graph for the last 4 hours, with a y-axis from 0 to 80 and an x-axis from 07:00 AM to 10:00 AM. The graph shows a constant value of approximately 88.12%. To the right of the graph, it indicates "4 Hours | 24 Hours | 7 Days" and "Minimum Value: 0", "Maximum Value: 100". Below the graph is an "Alert Details" section with a text box containing "SQL Server instance CH-SP2010-1 - Disk C is 88.12% full.". At the bottom, there are three buttons: "SHOW CATEGORY DETAILS", "SHOW INSTANCE DETAILS", and "CLOSE".

On the Alerts Detail screen you can find the following information:

- Alert Name
- Instance Name
- Value
- Current alert time
- Warning Threshold
- Critical Threshold
- Alert Details

Additional button options

Category

Click the **SHOW CATEGORY DETAILS** button to access first the Overview tab of a single instance and then specific category views such as sessions, queries, query waits, resources, databases, and alerts.

Instance

Click the **SHOW INSTANCE DETAILS** button to access the single instance dashboard for a specific instance.

Close

Click the **CLOSE** button to exit the Alerts Detail screen.

Prev

Click the **PREV** button to view details for a previous alert.


Next

Click the **NEXT** button to view details for the next alert.

Metric history

View alert history for the past 4 hrs, 24 hrs or 7 Days.

Launch SQL Diagnostic Manager

Click the SQL Diagnostic Manager icon  to launch the SQL Diagnostic Manager Console.

Access the Alerts Detail screen

SQL Diagnostic Manager provides several paths to access the Alerts Detail screen. The first access path is by clicking any row in the alerts roll up view of the Alerts tab. The second access path is by clicking over a category icon in the thumbnail sub-view that contains active alerts. It is also possible to access the Alerts Detail screen through the SQL Diagnostic Manager web console's home tab and the specific widgets it contains.

Advanced Filter

Advanced filters allows you to create custom filters that helps you segment instances using single or multiple conditions.

Advanced Filter dialog

- **Custom Filter Name** - Enter a name for your custom filter.

Use the following fields to create your custom filters. Custom filters on IDERA SQL Diagnostic Manager have the following structure:

- **Add Filter** - Allow you to add an additional filter.
- **Clear Filters** - Clears the value of filter creation fields.
- **Cancel** - Cancel the creation of the filter.
- **Apply Filters without Saving** - Applies the current advanced filter without saving it.
- **Save and Filter** - Saves and applies the filter.

Creating an Advanced Filter

To create an advanced filter follow these instructions:

1. Go to the instance view where you want to create a filter.
2. Click **Advanced Filter** and select **Create Filter**.

The screenshot shows the 'Alerts' section of the SQL Diagnostic Manager interface. At the top, there are tabs for 'Active', 'Severity', 'Instance', 'Metric', 'Category', 'SQLdmRepository', and 'Advanced Filter'. The 'Advanced Filter' tab is selected, and a dropdown menu is open with 'Create New Filter' highlighted. Below the menu, there are three alert entries, each with a red 'x' icon and a timestamp of '4/16/2019 1:11 PM'. The alerts are: 'aDashboardRepository is not backed up for 9999999 days.', 'Database ReportServer is not backed up for 9999999 days.', and 'Database model is not backed up for 9999999 days.'

3. On the Advanced Filters dialog enter a name for your filter.

The screenshot shows the 'Custom Filter Name' dialog box. It has a text input field with the value 'Filter_1' and a 'Help' link on the right side.

4. Use the drop down menus to select your desired criteria on the filter creation field.

The screenshot shows the filter creation field. It has a dropdown menu for 'Metric' set to 'Contains', and a list of criteria including 'SQL Server CP...' and 'Unsubscribed ...'. The logical operator is set to 'AND'.

5. Click **Save and Filter**.

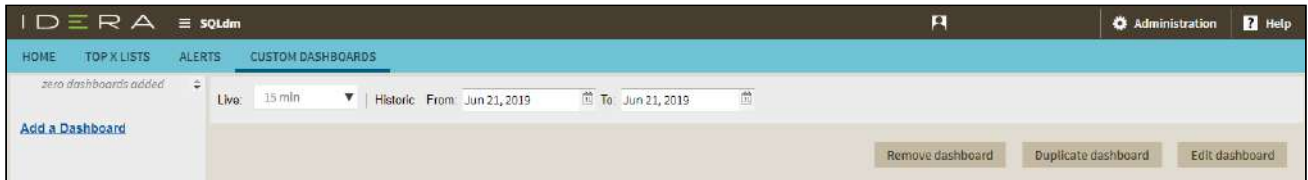
You can also use conditions to further refine your advanced filters. To add more filtering fields, click **Add Filter**.

The screenshot shows the filter creation field with two conditions. The first condition is 'Metric' set to 'Contains' with criteria 'SQL Server CP...' and 'Unsubscribed ...'. The second condition is 'Severity' set to 'Contains' with criteria 'Ok' and 'Informational'.

To remove a filter field click .

Create custom dashboards

The Custom Dashboards tab of the SQL Diagnostic Manager web console allows you to create and view custom dashboards. You can define and save multiple dashboard views to compare metrics across your monitored SQL Server instances.



Add a dashboard

To add a dashboard:

1. Click **Add a Dashboard** in the Custom Dashboards view.
2. Type a dashboard name on the Add a Dashboard window.
3. Select the Default check box, if you want to set this dashboard as the default one.
4. Click **Save**.

Add widgets to your dashboard

Customize your dashboard by adding widgets to display specific SQL Diagnostic Manager metrics. You can add a widget at a time.

To add a widget to your dashboard:

1. Click **Add a Widget** in the Edit Dashboard view.
2. Type a label for your widget in the corresponding box or leave it empty for default.
3. Select a widget type.

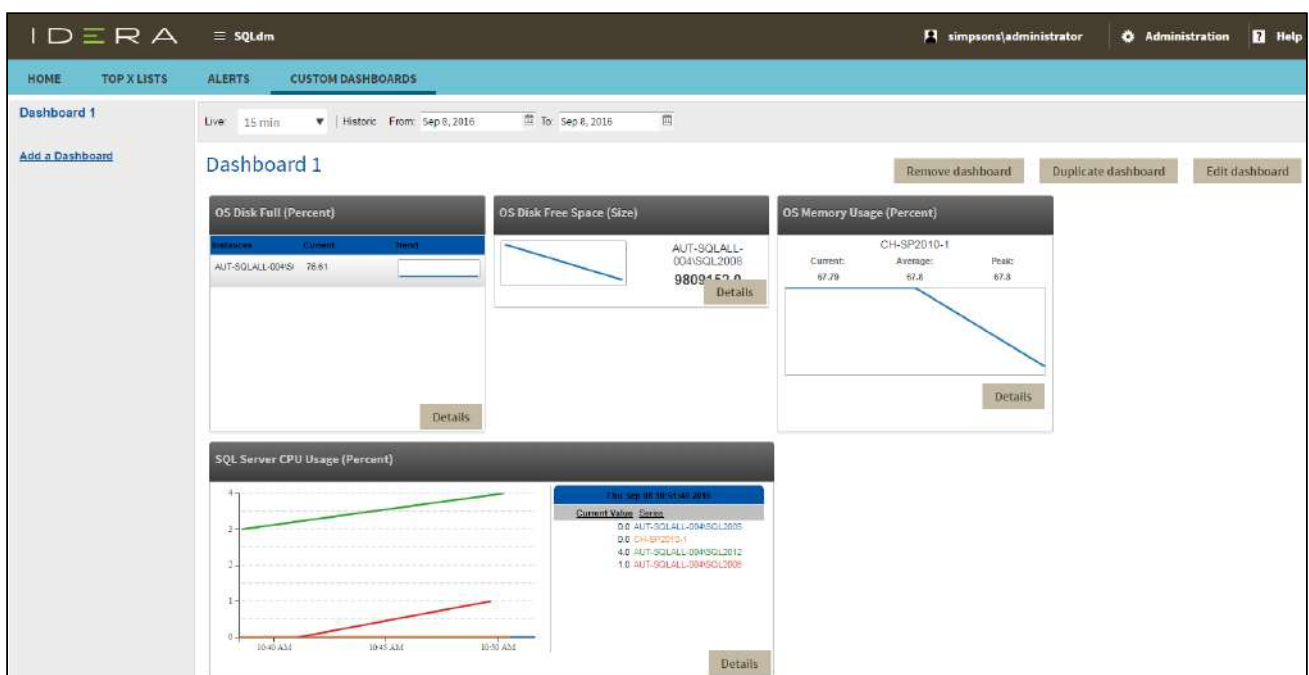
4. Choose a metric from the drop-down list.
5. Choose a data source from the drop-down list. Available options vary according to the type of widget selected and include:
 - Single Instance
 - Multiple Instance
 - Tag Name

- All Instances
6. Choose an identifier or identifiers from the drop-down list. Note that if you select the All Instances data source option, it is not possible to assign an identifier.
 7. Click **Save**.

Actions to perform in your Custom Dashboards View

The SQL Diagnostic Manager web console allows you to perform the following actions in your Custom Dashboards view:

- Filter your results by time frame
- Edit your dashboard
- Remove your dashboard
- Duplicate your dashboard



✔ To display a widget in full size, click **Details**.

Filter results by time frame

Filter results in your dashboard by specific built-in periods from the Live list or by specific start and end dates in the historic timeframe option.

⚠ By default, the results timeframe is the current date.


Edit your dashboard

In the Edit Dashboard view, you can:

- Change your dashboard name.

- Set your dashboard as the default one by selecting the Default check box.
- Select or create a dashboard tag by typing a name in the Dashboard Tags box.
- Add a new widget, remove, or edit current widgets.
- Place and order widgets in your dashboard by dragging and dropping them.

To access to the Edit Dashboard view, click **Edit Dashboard** in the Custom Dashboards view.

 Click **Save** after making changes in the Edit Dashboard view.

Remove a dashboard

To remove a dashboard, click **Remove Dashboard** and then select **OK** on the Confirmation window.

Duplicate a dashboard

To duplicate a dashboard, click **Duplicate Dashboard**.

Custom dashboards widget types

The SQL Diagnostic Manager web console allows you to choose between the following predefined widget types for a customized dashboard view:

Single value

Shows the single value for a specific metric and instance.

Mini-timeline

Displays the timeline and value for a specific metric and instance.

Value and timeline

Displays the value and the baseline range for a specific metric and instance.

List of Values

Displays the current value and the trend for a specific metric and instance, multiple instances, or tags.

Large timeline

Displays the timeline and the current value for a specific metric and instance, multiple instances, or tags.

Navigate the IDERA Newsfeed and SQLDM Mobile

The IDERA Newsfeed allows DBAs and managers to collaborate, share knowledge, and keep close tabs on their most critical SQL Server issues. Using real-time feeds, your team can increase productivity by sharing knowledge and expertise and staying up-to-date on the most recent issues.

Using Newsfeed, you can:

- Collaborate with team members regardless of where you or they are located.
- Follow application alerts, server activity, and coworker comments.
- Provide status updates as the team manages and resolves any issues.

With IDERA Newsfeed, you and your team stay informed, is more productive, and can shorten the time to problem resolution. You can access the Newsfeed user interface from either the SQLDM Console or the SQLDM Mobile Web application.

SQLDM Mobile helps you quickly investigate, diagnose, and correct performance and availability problems from your mobile device. This unique Web-based application, coupled with the IDERA Newsfeed technology, helps you stay productive and connected no matter where you or your servers are located.

With SQLDM Mobile, you can:

- See your entire SQL Server environment while “on the go”
- View real-time dashboards to identify problem areas and get current server health
- Drill down into the details of key performance metrics and alerts
- Collaborate with team members as issues unfold
- Use any mobile device or tablet

With SQLDM real-time performance data at your fingertips, you can keep your SQL Server enterprise running smoothly from any location.


Using the SQLDM Mobile & Newsfeed Service, SQLDM publishes status updates for each monitored server. These updates display on the server wall and in your Most Recent feed. You can interact with each update, posting troubleshooting information and brainstorming with coworkers. SQLDM stores all updates and their activity as stories in the SQLDM Mobile Repository.


Next, you can access to the following information on how to start using these services:

- [Configure your Newsfeed and Mobile deployment](#)
- [Create an account and Sign Up](#)
- [IDERA Newsfeed and SQLDM Mobile Log On](#)
- [Navigate the IDERA Newsfeed Interface](#)
- [Navigate SQLDM Mobile Dashboards](#)
- [Customize your experience](#)
- [Tips and tricks](#)


Configure your Newsfeed and Mobile deployment

After initially installing and setting up the IDERA Newsfeed and SQLDM Mobile, there are several tasks you can perform to further customize and streamline your deployment.

 For details on the supported installation scenarios, components and architecture, system requirements, and installation instructions see [IDERA Newsfeed and SQLDM Mobile](#).

 To deploy only the Newsfeed, see [Install IDERA Newsfeed only](#).

Suggested configuration workflow

	Follow these steps ...
	Ensure each SQLDM application has an owner.
	Verify that each SQL Server instance monitored by your SQLDM applications is available through the Newsfeed user interface. The IDERA Newsfeed Platform automatically creates a profile for each monitored instance so that your team can immediately begin following them.
	For each server: <ul style="list-style-type: none"> • Ensure it has an owner. • Change the default picture to a unique graphic or photo that correlates with the server name or can be easily identified as that server.
	Verify that the Repository grooming settings meet your data retention needs.

Available configuration tasks

- [Assign an owner to a server](#)
- [Assign an owner to an application](#)
- [Change a server or application profile](#)
- [Change the Web Server settings for SQLDM Mobile](#)
- [Change your account settings](#)
- [Change the Newsfeed administrator](#)
- [Configure how the IDERA Newsfeed Platform authenticates with SQLDM](#)
- [Configure the IDERA Newsfeed Repository](#)
- [Configure Mobile and Newsfeed email settings](#)
- [Invite coworkers](#)
- [Manage Users](#)
- [Manage your profile](#)
- [View application status](#)

Assign an owner to a server

Assigning an owner to a server provides a single point of contact for coworkers to call when that instance is experiencing performance issues. Server ownership can correlate with specific job responsibilities for monitoring your SQL Server environment.

To assign an owner using the SQLDM Console, follow these steps:

1. In the Newsfeed navigation pane, click **SQL Servers**.
2. In the right pane, select the server whose owner you want to change.
3. In the profile pane, click **Info**.
4. In the right pane, click **Edit**.
5. Use the **Owner** pull-down menu to select the appropriate person.
6. Click **OK**.

What happens when I assign server ownership to someone?

When you assign an owner to a server, the IDERA Newsfeed Platform automatically configures this user to follow the selected instance, ensuring them to easily and quickly access the server status updates from the Most Recent feed. The platform also updates the server profile to include the owner's contact information. After these updates are complete, the SQLDM Mobile and Newsfeed Service sends a notification email to the previous owner stating that the ownership has changed.

What are the benefits of server ownership?

Server ownership provides several benefits:

- **Following** - By owning a server, you automatically follow it.
- **Notifications** - You receive notifications from SQLDM Mobile & Newsfeed Service when anyone interacts with a server you own. To receive these notifications, change your account settings.

Who can set server ownership?

Any IDERA Newsfeed user who has SQLDM administrator rights can set ownership. To check these security settings, identify which SQLDM installation the IDERA Newsfeed Platform is using as the authentication provider, and then use the SQLDM Console to verify its security settings.

Assign an owner to an application

Assigning an owner to an application provides a single point of contact for coworkers to call when SQLDM is experiencing issues. Application ownership can correlate with specific job responsibilities for administrating your SQLDM installations.

To assign an owner using the SQLDM Console:

1. In the Newsfeed navigation pane, click **Applications**.
2. In the right pane, select the application whose owner you want to change.
3. In the profile pane, click **Info**.
4. In the right pane, click **Edit**.
5. Use the Owner pull-down menu to select the appropriate person.
6. Click **OK**.

What happens when I assign application ownership to someone?

When you assign an owner to an application, the IDERA Newsfeed Platform automatically configures this user to follow the selected application, ensuring an easily and quickly access to the application status updates from the Most Recent feed. The platform also updates the application profile to include the contact information.

What are the benefits of application ownership?

Application ownership provides several benefits:

- **Following** - By owning an application, you automatically follow it.
- **Notifications** - You can receive notifications from the SQLDM Mobile and Newsfeed Service about application activity. To receive these notifications, change your account settings.

Who can set application ownership?

Any IDERA Newsfeed user who has SQLDM administrator rights can set ownership. To check these security settings, identify which SQLDM installation the IDERA Newsfeed Platform is using as the authentication provider, and then use the SQLDM Console to verify its security settings.

Change a server or an application profile

You can change the server or application profile information that the Newsfeed displays .

To change profile settings using the SQLDM Console, follow these steps:

1. In the Newsfeed navigation pane, click either **SQL Servers** or **Applications**.
2. In the right pane, select the server or application whose profile you want to change.
3. In the profile pane, click **Info**.
4. In the right pane, click **Edit**.
5. Change the appropriate settings.
6. Click **OK**.

Change the Web Server settings for SQLDM Mobile

You can change the website address and port number for the SQLDM Mobile website or specify a different host computer for the IDERA Newsfeed Platform (Repository database and SQLDM Mobile & Newsfeed Service).

To specify new Web Server settings:

1. Start the SQLDM Web Manager console.
2. Specify the new website address or the name of the new host computer. When typing the website address, keep in mind the domain relationship between the IIS Server and your corporate network.
 - URL: PortNumber, as in *http://LABSERVER:85*, if the new website uses a port other than 80.
3. To ensure that SQLDM Mobile can connect to the new platform host, click **Test**.
4. Click **Save**.

Change your account settings

You can change the settings associated with your IDERA Newsfeed account at any time. Use the General Settings and Notification Settings views to update your account information or tweak your current activity notifications.

In the General Settings view, you can perform the following actions:

- Update your real name. Each Newsfeed account requires a unique, real name.
- Change the profile name that coworkers use to mention you in posts or comments using SQLDM Mobile.
- Specify a different email address. You will use this email address to log in to the Newsfeed and SQLDM Mobile. The SQLDM Mobile & Newsfeed Service uses this email address to send you notifications.

Access your account settings

In the Newsfeed menu bar of the SQLDM Console, click **Account**, and then click **Account Settings**. To edit your account, click **Change** located next to the setting you want to update.

Change the Newsfeed administrator

By default, the IDERA Newsfeed does not have an administrator. Consider choosing the same person who is responsible for administrating your SQLDM installation.

The Newsfeed administrator should be someone who can be contacted when important issues occur within the Newsfeed deployment, such as SQLDM Mobile & Newsfeed losing the connection to the SQLDM Mobile Repository database. When you specify the Newsfeed administrator, you provide the name, email, and work phone or mobile phone.

Follow these steps to set a Newsfeed administrator:

1. Start the IDERA Newsfeed Platform Manager console.
2. Click the **Administrator** tab.
3. Specify the appropriate settings.
4. Click **Save**.

Configure how the IDERA Newsfeed Platform authenticates with SQLDM

The IDERA Newsfeed Platform leverages the application security settings provided by SQLDM to authenticate users when they log on to the Newsfeed or SQLDM Mobile. You can specify a different authentication provider or change the port that the SQLDM Mobile & Newsfeed Service uses to communicate with the SQLDM Repository database.

✔ By default, the SQLDM Mobile & Newsfeed Service uses port 5166.

To specify new authentication provider settings:

1. Start the IDERA Newsfeed Platform Manager console, and navigate to the Authentication Provider tab.
2. Specify the name or IP address of the new host computer for the SQLDM Management Service. By default, the Management Service is installed on the computer where you ran the SQLDM setup program.
3. Verify that the selected port is open.
4. To ensure that the SQLDM Mobile & Newsfeed Service can connect to the new host, click Test.
5. Click Save.

When should I change the authentication provider settings?

You should specify new authentication provider settings when:

- You have migrated the SQLDM services to another computer.
- You want the IDERA Newsfeed to publish status updates from a set of SQL Server instances that are monitored by a different SQLDM installation.
- Your network security configuration has changed and you need to specify a different port.

Configure the IDERA Newsfeed Repository

The IDERA Newsfeed Platform lets you pick the Repository settings that best fit your environment and monitoring needs.

You can perform the following modifications:

- The Windows account or SQL login credentials that the SQLDM & Newsfeed Service uses to connect to the Repository database.
- The age limit at which the Repository grooming job permanently deletes stale activity and deleted or disabled objects.

If you have migrated the Repository to a new SQL Server instance, you can also change the name and location of the Repository database.

Change the grooming settings

You can set the age limits at which activity and unused objects are deleted from the Repository. For each grooming option, specify an age from 1 to 365 days.

To set your grooming options, follow these steps:

1. Start the IDERA Newsfeed Platform Manager console.
2. Click the **Repository** tab.
3. Under **Grooming Options**, select the application age limits for activity and objects. You can specify an age from 1 to 365 days.
4. Click **Save**.

The setup program installs a single grooming job that is scheduled to run on the Repository at midnight each day. This job calculates the age of all Newsfeed activity and objects stored in the Repository, and detects whether any instances have been removed from SQLDM. When the job finds stale active or disabled objects that are older than the specified age limit, it deletes the corresponding data, including all associated posts, status updates, comments, and bookmarks.

What gets groomed?

By default, the IDERA Newsfeed Platform grooms all unused objects and stale activity. You can change many of the age limits used to determine when to groom a specific object or activity.

- Applications that have been disabled for more than 45 days.
- Notifications older than 7 days.
- Newsfeed user accounts that have been disabled for more than 45 days.
- Servers that have not been monitored by the SQLDM application for more than 45 days.
- Stories older than 30 days.

- ✔ If an older story has had recent activity, such as a new comment, within the specified age limit, the IDERA Newsfeed Platform does not groom the story.

How often should I groom data?

How often you groom data — that is, how old you allow stories to get before deleting them from the SQLDM Mobile Repository — depends on these factors:

Factor	Why	How
The number of status updates that have been published by the SQLDM application	Based on this publication rate, you can determine how quickly the Repository database is growing to accommodate these stories, and how soon you will need to delete older stories to make room for future stories.	Use the Applications tab of the IDERA Newsfeed Platform Manager console to check the publication rate for your SQLDM application.
The amount of activity the average story gets	Because important issues typically receive the most attention, and get the most activity, you can gauge how long a story continues before it is considered "stale" by your team.	Use the activity timeline to track bursts of activity and identify how many stories are responsible for these peaks.

Factor	Why	How
Life-cycle of the average issue	The age limit at which you groom a story should be set to 2-5 business days from the point at which an issue is typically resolved, allowing ample time for your team to see the issue, respond to it, update the story, and note the solution.	Follow a few typical stories to see how they evolve and review your team's current issue resolution workflows.

✔ To keep an ongoing record of how key critical issues were solved, encourage your team to bookmark the corresponding stories, ensuring they will always have access to that information.

Change the IDERA Newsfeed and SQLDM Mobile service credentials

You can specify a different Windows account or SQL login for the SQLDM Mobile and Newsfeed service to use when connecting to the Repository database. Ensure that the account or the login meets the permissions requirements.

To specify new SQLDM Mobile & Newsfeed Service credentials:

1. Start the IDERA Newsfeed Platform Manager console.
2. Click the **Repository** tab.
3. Under Connection, input the following settings:
 - Choose the type of authentication.
 - Specify the user name (for a Windows account, type domain\username).
 - Specify the password.
4. Click **Save**.

Configure SQLDM Mobile and Newsfeed email settings

In order to receive notifications from the SQLDM Mobile and the Newsfeed service, you must configure how the IDERA Newsfeed Platform should communicate with your Mail Server.

Available email settings

In the Email Settings you can perform the following changes:

- Name of your Mail Server.
- Port that the SQLDM Mobile & Newsfeed Service should use to communicate with the Mail Server (by default, port 25).
- How long (in seconds) the IDERA Newsfeed Platform should wait for a reply from the Mail Server before timing out (by default, 15 seconds).
- Whether the Mail Server requires authentication, and which credentials should be used for authentication.
- The name and the email address that the SQLDM Mobile & Newsfeed Service should use when sending notification emails.

To configure your email settings, follow these steps:

1. Start the IDERA Newsfeed Platform Manager console.
2. Click the **Notification** Server tab.
3. Specify the appropriate values.
4. To ensure SQLDM Mobile can connect to your Mail Server, click **Test**.
5. Click **Save**.

Invite coworkers

Manage users

In the option Manage Users, you can delete, disable, or enable a Newsfeed user account.

Delete a user account

Deleting a user account prevents the affected team member from logging on to either the IDERA Newsfeed or SQLDM Mobile, and from receiving notification emails about server or coworker activity.

The disabled user is also queued for grooming. Depending on your settings, the grooming job permanently deletes the account configuration, profile, and status update information from the Repository after the selected number of days have passed.

If you enable the user before it is groomed, the IDERA Newsfeed Platform restores the account to its most recent state. Once a user account is groomed, it cannot be re-instated. To temporarily prevent a team member from using the IDERA Newsfeed or SQLDM Mobile, consider disabling the account instead.

You may need to delete a user when a team member has moved to another department or left the company and when someone has violated the policies of the company.

Follow these steps to delete a user account:

1. Start the IDERA Newsfeed Platform Manager console.
2. Click the **Users** tab.
3. From the list of registered users, select the user you want to delete.
4. Click **Delete**.

Disable a user account

Disabling a user account prevents the affected team member from logging on to either the IDERA Newsfeed or SQLDM Mobile, and from receiving notification emails about server or coworker activity.

You may need to disable a user when you are setting up and configuring the IDERA Newsfeed deployment. Also, when a user has violated the company policies.

Follow these steps to disable a user account:

1. Start the IDERA Newsfeed Platform Manager console.
2. Click the **Users** tab.
3. From the list of registered users, select the user you want to disable.
4. Click **Disable**.

Enable a user account

Enabling a Newsfeed user allows that person to receive status updates, alerts, and other pertinent performance data about monitored SQL Server instances in your environment. When a user is enabled, your team can interact with this coworker and receive notifications about their activity.

By default, an account and profile are created for a user when they sign up with the IDERA Newsfeed. All new users are automatically enabled, and can begin monitoring servers.

Follow these steps to enable a user account:

1. Start the IDERA Newsfeed Platform Manager console.
2. Click the **Users** tab.
3. From the list of registered users, select the user you want to enable.

4. Click **Enable**.

Manage your profile

Manage your profile by checking activity on your wall and keeping your profile information up-to-date to ensure your coworkers can easily find you and interact with you.

To check your wall activity using the SQLDM Console, follow these steps:

1. On the Newsfeed menu bar, click **Profile**.
2. In the profile pane, click **Wall**.
3. In your wall, you can perform the following actions:
 - Search for a specific story.
 - Use the activity timeline to track communications between you and your coworkers.
 - Review the details of a story to see how a conversation resolved.
 - Like, bookmark, or comment on a post.
4. Click **OK**.

To check your wall activity using SQLDM Mobile, follow these steps:

1. On the SQLDM Mobile home page, tap **Profile**.
2. From your wall, you can perform the following actions:
 - Review the details of a story to see how a conversation resolved.
 - Like, bookmark, or comment on a post.
3. Use the posts filter to show all posts or only your shared posts.

To change your profile using the SQLDM Console, follow these steps:

1. On the Newsfeed menu bar, click **Profile**.
2. In the profile pane, click **Info**.
3. In the right pane, click **Edit**.
4. Change the appropriate setting.
5. Click **OK**.

View application status

In the IDERA Newsfeed Platform Manager, you can check the status and other properties of the SQLDM application you registered with the IDERA Newsfeed.

You can view the following properties:

- The date and time of last published status update.
- The total number of status updates that have been published by SQLDM.
- The number of servers being monitored by SQLDM.
- The number of coworkers who are following this application.

Follow these steps, to view your application status:

1. Start the IDERA Newsfeed Platform Manager console.
2. Click the **Applications** tab.
3. To ensure the displayed properties are current, click **Refresh**.

Create an account and Sign Up

IDERA Newsfeed lets you take advantage of its monitoring and collaboration features from within the SQLDM Console or the SQLDM Mobile application on your mobile device. The signup process includes creating your Newsfeed account and completing your profile.

Both the IDERA Newsfeed feature and SQLDM Mobile use your Newsfeed account for login authentication. After you have signed up, you can immediately begin interacting with coworkers and checking your servers per your SQLDM application security level. To better understand your security level, contact your SQLDM administrator.

- ✓ When you sign up with the IDERA Newsfeed, your account is automatically set up to follow your own posts. This inclusion allows you to see your posts in the context of the Most Recent feed, just as your posts appear to your coworkers.

Sign up via the SQLDM Console

Use the SQLDM Console to sign up with the IDERA Newsfeed.

1. In the navigation pane of the SQLDM Console, click **Newsfeed**.
2. Under Sign Up in the right pane, type the following information:
 - Your first and last name.
 - The email address you want the SQLDM Mobile & Newsfeed Service to use when sending you notifications.
 - The nickname or Profile Name you want your coworkers to use when mentioning you in their posts or comments.
3. Click **Sign Up**.
4. On the Complete Your Profile window, specify your current title and preferred contact information, and then click **Save & Continue**.

- ✓ You can also upload a photo to help your coworkers easily identify you in their feeds or find you when searching for your comments and posts.

5. On the Follow Servers window, choose which monitored SQL Server instances you want to follow, and then click **Save & Continue**.
6. On the Follow Coworkers window, choose which coworkers you want to follow, and then click **Save & Continue**.
7. The Newsfeed pane displays the Welcome window, from which you can complete additional customizations.

Sign up via SQLDM Mobile

You can sign up and create your Newsfeed account from the SQLDM Mobile Web application.

1. Using your desktop or mobile Internet browser, go to the SQLDM Mobile Web application. The default Web address or URL for SQLDM Mobile is: `//MyWebServer/SQLdm`.
2. On the login view of SQLDM Mobile, click **Sign Up**.
3. On the Sign Up view, type the following information:
 - Your first and last name.
 - Your current title and preferred contact information.
 - The email address you want the SQLDM Mobile & Newsfeed Service to use when sending you notifications.

- The nickname or Profile Name you want your coworker to use when mentioning you in their posts or comments.
 - The credentials you want to use for your Newsfeed account.
4. Click **Log On**.
 5. In the servers view, choose which monitored SQL Server instances you want to follow, and click **OK**.
 6. In the coworkers view, choose which coworkers you want to follow, and click **OK**.
 7. SQLDM Mobile creates your Newsfeed account and then displays its home page.

IDERA Newsfeed and SQLDM Mobile Log On

You can log on to the IDERA Newsfeed from within the SQLDM Console or to SQLDM Mobile from your mobile device. The default Web address or URL for SQLDM Mobile is: `//MyWebServer/SQLdm`.

When you log on to SQLDM Mobile, the SQLDM Management Service validates your email address against the address specified for your Newsfeed account and then validates your password against the credentials of your SQLDM account. SQLDM Mobile does not store your password. In environments that use Windows authentication, SQLDM Mobile impersonates your user account when connecting to the SQLDM Repository database. In environments that use SQL Authentication, SQLDM Mobile uses the provided username and password. When the Repository connection succeeds, the SQLDM Mobile Log In is authenticated and remains in use until you log off from SQLDM Mobile.

- ✔ To get a more streamlined experience with the IDERA Newsfeed, such as directly linking to an ongoing story from a notification email, choose to keep yourself logged on.

To log on to the IDERA Newsfeed within SQLDM:

1. In the navigation pane of the SQLDM Console, click **Newsfeed**.
2. In the Email text box at the upper right corner, type the email address you specified when you signed up with the IDERA Newsfeed.
3. Click **Log In**.

To log on to SQLDM Mobile using your Mobile device:

1. If you want to connect through a VPN:
 - a. Connect to your corporate network or Intranet using your Windows credentials.
 - b. Type the SQLDM Mobile URL in the address bar of your mobile browser.
2. If you want to connect without a VPN, type the SQLDM Mobile URL in the address bar of your mobile browser.
3. On the Log On window, type the:
 - Email address you specified when you signed up with the IDERA Newsfeed.
 - Password of the Windows account you use when accessing SQLDM.
4. Tap **Log In**.

Navigate the IDERA Newsfeed Interface

The IDERA Newsfeed is a new way for DBAs and managers to collaborate, share knowledge, and keep close tabs on your most critical SQL Server issues.

In the following list, you can find some basic components that can help you understand more about the terminology used in IDERA Newsfeed:

- [Activity Timeline](#)
- [Application](#)
- [Coworker](#)
- [Comment](#)
- [Followers](#)
- [Like or Unlike](#)
- [Mention](#)
- [My Bookmarks](#)
- [My Directory](#)
- [My Favorites](#)
- [Post](#)
- [Profiles](#)
- [Server](#)
- [Status Update](#)
- [Story](#)
- [Walls](#)

Activity Timeline

An activity timeline is a chronological graph that displays in the SQLDM Console, at the top of your Most Recent feed as well as on the walls of servers, your SQLDM application, and coworkers. It shows how many stories have been created over time, as well as when stories have caused the most interaction, allowing you to quickly identify the highest and lowest points of activity.

How can the activity timeline help me diagnose issues?

By identifying when activity spikes happened, you can discern when and where you should focus your attention. You can then drill down to the corresponding stories that occurred at specific points in time by selecting suspicious spikes in the timeline.

The activity timeline also lets you easily discover historical patterns, such as a repeating server issue, or return to a past event to see how it was resolved.

How do I browse activity on the timeline?

Choose whether you want to browse activity by day, month, or year, and then use the forward and backward arrows to scroll through the corresponding timeline.

How do I pick a specific point in time?

When you hover over or tap on the activity timeline graph, a horizontal bar displays, pinpointing the exact point in time for which you are seeing activity. To change this point in time, select the bar and move it to the appropriate time.

Application

An application is a tool, such as SQL Diagnostic Manager, that provides alerts and other event information about servers to the IDERA Newsfeed. This information is processed into status updates in your feed and on the wall of your server. To better understand the relationship between an application and its managed servers, learn how the IDERA Newsfeed works.

How do I see a list of all available applications?

In the Newsfeed navigation pane of the SQLDM Console, click **Applications**.

How do I see information about my application?

You can view information about your SQLDM application from the profile of an application or the Applications tab in the IDERA Newsfeed Platform Manager console.

From the info tab of the profile, you can see general information such as:

- The profile name of the application, which you can use to mention the application in your posts and comments.
- Who is responsible for administrating this application, the application owner.
- Who is following this application, the followers.

From the Applications tab in the IDERA Newsfeed Platform Manager console, you can view statistics such as:

- Date and time of the last published status update, the recent activity.
- The total number of updates published since install.
- The number of SQL Server instances monitored by this SQLDM application.

How do I get to a wall of an application?

In the Newsfeed navigation pane of the SQLDM Console, click **Applications**, and then click the name of the SQLDM application whose wall you want to view. You do not have to follow the application to view its wall.

How do I follow an application?

In the Newsfeed navigation pane of the SQLDM Console, click **Applications**, and then click Follow for each SQLDM application you want to follow.

Why should I follow an application?

Some applications, such as SQLDM, provide status updates and system alerts about the health of their components. For example, SQLDM will alert you when it detects an issue that involves the Collection Service or if the Repository was recently offline. Following an application lets you track these status updates from your feed.

Comment

Comments allow you to interact with an active story. Use comments for the following actions:

- Update the status of a server to provide new troubleshooting data or confirm a previously applied fix.
- Assign a task to your coworker.
- Ask your coworkers questions about an application alert.

To add your comment to a story, select **Comment** in the story summary.

Whenever you click Like in a comment, your coworker is notified giving them the opportunity to respond.

Coworker

A coworker is a teammate, manager, or coworker who has signed up with the IDERA Newsfeed. You can send invitations to anyone inside your SQL Server organization who also uses SQL Diagnostic Manager.

How do I see a list of all my coworkers?

In the Newsfeed navigation pane of the SQLDM Console, click **Coworkers**.

Does it matter where my coworkers are located?

No. Coworkers can be located in any geographic location, and own servers that you are responsible for monitoring.

How can I interact with my coworkers?

You can interact with your coworkers by:

- Follow coworkers so you can see their posts in your Most Recent feed.
- Ask your coworker a question by posting on their wall.
- Comment on posts shared by your coworkers.
- Reply to comments coworkers have posted on stories.
- Keep up-to-date when your coworker profile contact information changes.
- Tag a coworker, mention them in a comment or post.

How do I get to a coworker's wall?

In the Newsfeed navigation pane of the SQLDM Console, click **Coworkers**, and then click the name of the coworker whose wall you want to view. You do not have to follow the coworker to view their wall.

How do I follow a coworker?

In the Newsfeed navigation pane of the SQLDM Console, click **Coworkers**, and then click **Follow** for each coworker you want to follow.

Why should I follow a coworker?

Follow a coworker if you want to see their posts and related activity in your feed.

Followers

Followers are those coworkers who are following you or the servers you own. In turn, you can follow any coworker, server, or application whose posts and status updates interest you. Following someone or something makes it easy to keep up with recent activity and collaborate on specific issues.

When you choose to follow someone or something, you automatically begin to see their posts and status updates in your Most Recent feed. From the feed, you can easily:

- Comment on posts and status updates from the people and servers you are following.
- Like the posts and comments shared by your coworkers.
- Bookmark stories originated by your coworker or by a server.

Each profile lists who is following that person or that server. You can always view stories, post questions, or post announcements to walls, even though you are not following them. You can choose to follow specific stories without following that coworker or server.

Why do I see my own posts in my feed?

When you sign up with the IDERA Newsfeed, your account is automatically set up to follow your own posts. This inclusion allows you to see your posts in the context of the Most Recent feed, just as your posts will appear to your coworkers.

How do I follow someone or something?

In the Newsfeed navigation pane of the SQLDM Console, click either **Coworkers, Application**, or **SQL Servers**. Then, click **Follow** located next to the name of the person or server you want to follow.

How do I see who is following me?

In the SQLDM Console, navigate to the Newsfeed, and click **See All** located next to **Followers** in the right pane. To go to a specific person's wall, click the appropriate thumbnail profile picture.

How do I review the things and people I am following?

From the SQLDM Console, navigate to the Newsfeed, and then click **See All** next to Following in the right pane. To go to a specific wall, click the appropriate thumbnail profile picture.

How do I follow specific stories?

You can follow specific stories shared by servers you are not following. You can also ignore specific stories shared by the servers you are following.

From the SQLDM Console, navigate to the wall of the server whose story you want to follow, find the story, and then click **Follow** in the story summary.

Like or Unlike

When you give a like to a post of a coworker you are following shares, it means that this information is important to you and you agree with the idea or the assessment.

To like a story, select **Like** in the story summary or tap **Like** on the action menu of the Wall view.

When you like a post, your coworker may be notified of your action, giving them the opportunity to respond. If you want to change your assessment later, you can select or tab **Unlike** the story.

Mention

You can tag a coworker or server in your post or comment by mentioning that person or thing. Mentioning a coworker lets you notify him or her of a task that needs to be done or a question you have. Mentioning a server lets you point out a problem, or draw correlations between similar events on different servers.

Mention someone or something can also have the following effects:

- Notifies your coworker that he or she has been mentioned, giving him or her the opportunity to respond.
- Notifies the owner of the server about your post or comment.
- Displays the corresponding story when coworkers search for posts or comments about a particular server or person.

To mention a coworker or server, type @ and then the name of the person or the name of the computer. The feed will automatically match the name to available coworkers and servers, refine the match criteria as you type. An

available coworker is any person who has a Newsfeed account whereas an available server is any server that is being managed by an application.

You can mention someone or something "inline", in the context of the comment or post you want to make.

My Bookmarks

A bookmark is a story that you consider important and you want to keep track closely as it evolves. Bookmarking helps you perform the following actions:

- Set specific stories aside for quick and easy access.
- Create a list of the top servers or issues you want to track today or this week.
- Keep a permanent copy of the selected story.

When you bookmark a story, it is copied to the Bookmarks view. The IDERA Newsfeed Platform does not groom bookmarked stories. Using bookmarks, you can keep an ongoing, historical record of mission-critical issues and their solutions.

You can bookmark any story displayed in your Most Recent feed or on the wall of a coworker or server you are following.

How do I bookmark a story?

To copy a story to your Bookmarks view, select **Bookmark** in the story summary. A copy of the story remains in your feed. Bookmarked stories are not groomed.

How do I use the Bookmarks view?

You can interact with your Bookmarks view in much the same way you interact with your feed. Your bookmarked stories are listed in chronological order, displaying the most recently updated story first.

You can scroll through your bookmarks, review updates, and add your comments. You can also remove a bookmark from this view by selecting **Remove Bookmark** in the story summary.

How do I get to the Bookmarks view?

In the Newsfeed navigation pane of the SQLDM Console, click **Bookmarks**.

How do I remove stories from the Bookmarks view?

To remove a story from your Bookmarks view, select **Remove Bookmark** in the story summary.

A copy of the story remains in your feed until the next grooming job starts.

My Directory

Your directory is a list of all the applications, servers, and coworkers registered with the IDERA Newsfeed. You can use the directory to go directly to the wall of the person or server you are following.

In the SQLDM Console, the directory is integrated into the Newsfeed pane. In SQLDM Mobile, the directory is a unique view. In either interface, you can browse the listings or search for the specific people and things you want to find.

How do I get to my directory?

In the Newsfeed navigation pane of the SQLDM Console, click SQL Servers, Coworkers, or Applications.

My Favorites

Favorites are those people and things you already follow but want easy access to their walls. By marking a coworker or server as a favorite, the IDERA Newsfeed Platform creates a shortcut to that person or thing so that you no longer need to search for them or use your directory to look them up.

In the SQLDM Console, shortcuts to your favorites are listed in Newsfeed navigation pane. In the SQLDM Mobile interface, shortcuts to your favorites are displayed on the SQLDM home page.

- ✔ Consider limiting your favorites list to important people and servers, such as your team members and the mission-critical instances for which you are responsible.

How do I get to my favorites?

In the Newsfeed navigation pane of the SQLDM Console, click the appropriate person or thing from your favorites list.

How do I change my favorites?

From the Newsfeed navigation pane of the SQLDM Console, click Edit Favorites, and then choose the appropriate action for each person or thing in the list.

My Most Recent feed

Your Most Recent feed is a real-time aggregation of stories initiated by the people and things you are following, including the latest interactions and updates. You can access this feed from both the SQLDM Console and SQLDM Mobile.

- ✔ You can declutter your feed by unfollowing a story that is no longer important to you.

How many stories can I read at one time?

In the SQLDM Console, the Most Recent feed displays a total of 200 stories, from most recent to oldest, in lists of 20 stories at a time. You can interact with a story from within the feed by clicking one of the actions or links available in the story summary.

What is the story summary?

The story summary provides a succinct synopsis of the story's evolution and activity, including the original post or status update, a count of how many comments and updates have been made so far, as well as links to more information and the actions you can take.

From most stories, you can:

- Like the story.
- Expand the comments and updates to see the full history of the story (in SQLDM Mobile, tap the story to see details).
- Change whether the story is bookmarked (in SQLDM Mobile, this is available from action menu on the Story details view).
- Comment the post or the story.

- Delete your comments.

How does the Most Recent feed work?

By default, the Most Recent feed lists all of the most recent posts and status updates shared by the people and things you are following. This feed represents the state of your enterprise as of now.

You can filter the Most Recent feed to focus on stories for people, applications, or servers. You can also use the activity timeline to focus the feed on what happened during a particular point in time.

How do I get to my feed?

Log in to IDERA Newsfeed from the SQLDM Console. By default, the Most Recent feed displays in the right pane.

My Notifications

Notifications keep you up-to-date on the latest activity initiated by the people and things you follow, even when you are not logged in to the IDERA Newsfeed or SQLDM Mobile.

You can control what information you get and how you get it. Edit your notification settings to choose which activities are important to you and whether you want to receive an email when the SQLDM Mobile & Newsfeed Service sends you the corresponding notification.

To customize which notifications you receive, click **Edit Settings** in My Notifications view or change your Account Settings.

When do I receive notifications?

You receive notifications based on the notification settings you have enabled. By default, the SQLDM Mobile & Newsfeed Service generates notifications for most types of activity. Each time the specified activity happens, the service notifies you in "real time" by updating the My Notifications view and sending you an email.

Note that several notification settings rely on you, or someone else, to interact with a post or status update.

How do I use the My Notifications view?

Use My Notifications view to review notifications that have accumulated over the last 7 days. This view summarizes your notifications per type of activity per day. Most notifications will include links to the servers and coworkers who initiated the activity.

How do I get an email notification?

In the Newsfeed navigation pane of the SQLDM Console, click **Notifications** and then click **Edit Settings**. On My Notification Settings view, enable email notifications.



By default, the SQLDM Mobile & Newsfeed Service is configured to send you email notifications. If you are not receiving notifications when you think you should, verify the IDERA Newsfeed Platform configuration.

How do I view notifications when I don't have email enabled?

In the Newsfeed navigation pane of the SQLDM Console, click **Notifications**.

Why am I getting 2 emails for each server alert?

Depending on your SQLDM alert configuration and your IDERA Newsfeed notification settings, you may receive more than one email that notifies you about a server alert. To avoid duplicate emails about the same alert notification, change one of these settings:

- Disable all email notifications.
- Deselect the Posts a status update for something you own notification setting for application activity.
- Modify the Send an email to Alert Response setting in SQLDM by either disabling this action or changing the target recipients.

Post

A post is an announcement or status update that is shared by a coworker, or shared by you about a coworker or server. All posts shared by coworkers you follow are displayed in chronological order, from most recent to oldest, in your Most Recent feed.

You can like, bookmark, or comment on any post in your feed. This interaction is considered activity. When a post has activity, it becomes a story that has a specific beginning (the original post) and end (the resolution). Previous stories and their activity can be tracked using the activity timeline available at the top of your feed and the wall of the coworker.

Profiles

Each profile consists of a wall and an information page, providing a centralized location that displays all the data about a specific person or thing.

Each server, application, and coworkers has a profile. From their profiles, you can:

- See who owns a particular server or application.
- Change the profile settings for an application or server.
- Look up an email address or a mobile phone number.
- Verify the name of an application.

Application Profile

An application profile consists of its wall and an information page, providing a centralized location that displays all the data about a specific application.

The application profile lets you:

- See who owns this application.
- Review stories and status updates posted to its wall.
- Change the profile settings for the application.
- Verify the name and owner of an application.

Coworker Profile

A coworker profile consists of its and an information page, providing a centralized location that displays all the data about a specific person with whom you work.

The coworker profile lets you:

- Review stories posted to his or her wall.
- Verify contact information, such as email addresses and phone numbers.
- Look up the coworker's profile name so you can mention him or her when using SQLDM Mobile.

My Profile

Your profile consists of your wall and an information page, providing a centralized location that displays all the data about you.

Your profile lets you:

- Review stories that you and your coworkers have posted on your wall.
- Change your profile picture.
- Update your contact information, such your mobile phone number.
- Update your company information.

To update other data, such as your email address or the profile name that coworkers use to mention you in SQLDM Mobile, change your account settings.

Server Profile

A server profile consists of its wall and an information page, providing a centralized location that displays all the data about a specific server.

The server profile lets you:

- See who owns this server.
- Review stories and status updates posted to its wall.
- Change the profile settings for the server.
- Verify the name of the server and owner.
- Check virtual machine properties (when applicable — this data is available in the SQLDM Console Newsfeed pane and in SQLDM Mobile on tablet devices).

Server

A server is a SQL Server instance that is monitored by a SQL Diagnostic Manager (SQLDM) application.

How do I see a list of all monitored instances?

In the Newsfeed navigation pane of the SQLDM Console, click SQL Servers.

Does it matter where my servers are located?

No. Servers can be located in any geographic location. Within the IDERA Newsfeed or SQLDM Mobile, you can access any server that you would normally be able to monitor using the SQLDM Console.

How can I interact with a server?

You can interact with your monitored servers by:

- Following servers so you can see their status updates in your Most Recent feed.
- Announcing (un)scheduled maintenance activities on the wall of the server so that coworkers know when the server will be unavailable.
- Commenting on server status updates to share your latest findings about an issue.
- Marking the server as a favorite so you can easily and quickly access it.
- Keeping up-to-date when the profile information of a server changes
- Tagging a server by mentioning it in a comment.

How do I get to a servers wall?

In the Newsfeed navigation pane of the SQLDM Console, click **SQL Servers**, and then click the name of the SQL Server instance whose wall you want to view. You do not have to follow the server to view its wall.

How do I follow a server?

In the Newsfeed navigation pane of the SQLDM Console, click **SQL Servers**, and then click **Follow** for each SQL Server instance you want to follow.

Why should I follow a server?

Follow a server if you want to see its status updates and related activity in your feed. To get notified about specific alerts and events concerning the server, assign yourself as the server owner.

Status Updates

A status update is an event, alert, or other message published by an application for a server it is managing. All status updates published by applications about servers you follow are displayed in chronological order, from most recent to oldest, in your Most Recent feed. As more information about the event or alert is collected, the application updates the original status.

You can bookmark or comment on any status update in your feed. This interaction is considered activity. When a status update has activity, it becomes a story. Previous stories and their activity can be tracked using the activity timeline available at the top of your feed.

Story

A story is a post or status update that is experiencing activity. Stories track the life cycle of a post, from its origin to the last activity. For example, when a coworker has commented on a post, or a server has a new status update about the same alert, those activities are added to the story and displayed in chronological order under the original post. As this activity continues, the story is updated to reflect its current state.

Stories display both in your Most Recent feed and on the corresponding server and coworker walls. In your feed, you will see only those stories originated by the applications, coworkers, and servers you are following. You can also choose to follow specific stories shared on the walls of coworkers and monitored servers you are not following.

You can like, comment on, or bookmark any post or server status update.

Over time, stories are groomed from the SQLDM Mobile Repository. You can configure the age limit at which to groom old stories.

Walls

Walls display chronological lists, from most recent to oldest, of shared posts and status updates.

Each server, application, and coworker has a wall. From their walls, you can:

- Share an announcement or ask a question.
- Find a specific story.
- Track activity bursts along the Activity Timeline.
- Follow or unfollow this person or thing.
- Add this person or thing to your favorites.
- See who is following this person or thing.
- See who and what your coworkers are following.
- Use Helpful Links to complete your profile or follow coworkers.

Application Walls

An application wall shows the activity for a specific application. Use the wall to scroll through a chronological list of status updates, from most recent to oldest, that have been shared by this application. You can also use the wall to communicate to anyone following the application.

The application wall lets you:

- Share an announcement or ask a question about this application.
- Find a specific story initiated by the application or someone following the application.
- Track bursts of application activity along the Activity Timeline.
- Follow or unfollow this application.
- Add this application to your favorites.
- See who is following this application.
- Use Helpful Links to complete other tasks.

Coworker Walls

A coworker wall shows the activity for a specific coworker. Use the wall to scroll through a chronological list of posts, from most recent to oldest, that have been shared by this person. You can also use the wall to communicate to anyone following this person.

The coworker wall lets you:

- Share an announcement or ask a question about this person.
- Follow or unfollow this person.
- Follow a story this person shared.
- Find a specific story shared by a specific person.
- Track bursts of activity about this person along the Activity Timeline.
- Add this person to your favorites.
- See who is following this person.
- See who and what this person follows.
- Use Helpful Links to invite other coworkers to join the conversation.

My Wall

Your wall shows the activity initiated by you. Use the wall to scroll through a chronological list of posts, from most recent to oldest, that have been shared by you and anyone who has posted to your wall. You can also use the wall to communicate to anyone following you.

Your wall lets you:

- Share an announcement or ask a question of your followers.
- Find a specific story initiated by you or a coworker who posted to your wall.
- Track your activity bursts along the Activity Timeline.
- See who is following you.
- Review who and what you are following.
- Use Helpful Links to invite other coworkers to join the conversation.

Server Wall

A server wall shows the activity for a specific server. Use the wall to scroll through a chronological list of status updates, from most recent to oldest, that have been shared by this server. You can also use the wall to communicate to anyone following the server.

The server wall lets you:

- Share an announcement or ask a question about this server.
- Find a specific story shared by the server or someone following the server.
- Follow or unfollow this server.
- Follow a story this server initiated.
- Unfollow a story, removing it from your Most Recent feed.
- Track bursts of server activity along the Activity Timeline.
- Add this server to your favorites.
- See who is following this server.
- Use Helpful Links to complete other tasks.

Navigate SQLDM Mobile Dashboards

SQLDM Mobile helps you diagnose and fix SQL Server performance issues at any time from your smart phone or other mobile devices.

Here, all the options that are available to you through SQLDM Mobile dashboards:

- [Active alerts](#)
- [Database details and databases dashboard](#)
- [Home Page](#)
- [Key resource metrics](#)
- [My servers](#)
- [SQLDM Dashboard](#)
- [Session](#)
- [SQL Server Agent](#)

Active alerts

SQLDM Mobile tracks the active alerts for the SQLDM operations you are monitoring. The Active Alerts view provides a summary of each active alert so you can easily triage the corresponding performance issues. The alert summary includes the alert name and description, as published by SQLDM.

To get to the Active Alerts view for each section, tap the Dashboard tab and tap the alert area you want to review. The following alerts are displayed this dashboard:

- Operational Active alerts.
- Queries Active alerts.
- Services Active alerts.
- Logs Active alerts.
- Custom Active alerts.

✔ All the active alerts displayed in the SQLDM Mobile are arrange by severity.

To see alert details, such as the affected SQLDM component and the complete alert message, tap the active alert for more information.

You can get the server profile view by tapping the **Dashboard** tab and then tap the status of active alert you want to explore.

Database details and databases dashboard

Database details

The Database view displays key performance metrics for the selected database, such as:

- Type of database (user or system) and its status
- Total number of sessions connected to this database
- Total number of user tables in this database
- Size of the database data file versus the amount of disk space available for it to grow, and the percentage of free space used
- Size of the database log file versus the amount of disk space available for it to grow, and the percentage of free space used
- When the oldest open transaction started

- When the last backup of this database occurred
- Any alerts that SQLDM has generated about this database

The Databases view lists all of the user and system databases hosted by the selected SQL Server instance, and provides a status summary for each database that includes the overall health and total number of connected sessions.

You can swipe this list to quickly assess the health of the server or search for a specific database whose performance concerns you.

To see more performance metrics about a database, tap the appropriate database in the list.

Database dashboard

The databases view lists all of the user and system databases hosted by the selected SQL Server instance, and provides a status summary for each database that includes the overall health and total number of connected sessions.

You can swipe this list to quickly assess the health of the server or search for a specific database whose performance concerns you.

To see more performance metrics about a database, tap the appropriate database in the list.

Home Page

The Home Page provides shortcuts to the key features available in SQLDM Mobile, as well as quick access to your favorite coworkers and servers.

To access to your home page, log in to SQLDM Mobile and tap **Home** on the action menu in the upper right hand corner of your screen.

In the Home Page you can perform the following actions:

- Assess the health of your servers.
- Get the latest updates from your Most Recent feed.
- View notifications about recent activity.
- Check for updates on bookmarked stories.
- Use the directory to discover which people and things are registered to SQLDM Mobile.
- Go to your wall to post a status update.
- See what is happening with your favorites.

Key resource metrics

The Resources view gives you access to the key metrics you need to monitor for optimal resource performance.

You can view the current value of these metrics or plot them from now up to 12 hours in the past. These charts can help you identify performance patterns or unexpected peaks in activity.

- 30 minute period, with a data point every 5 minutes.
- 1 hour period, with a data point every 10 minutes.
- 12 hour period, with a data point at each refresh interval (by default, every 6 minutes).



Some resource metrics are OS metrics. To successfully monitor OS metrics, ensure OLE Automation is enabled on the monitored SQL Server instance and in the SQLDM Console.

Summary

The Summary filter presents key metrics representing the most important performance indicators across all resources. When this server is a virtual machine (VM), the Summary filter displays the CPU and memory usage consumed by the VM workload.

- Response Time (in milliseconds)
- CPU Usage
 - SQL Server usage (percentage)
 - Total usage (percentage)
- Memory Usage
 - Used by SQL Server (in kilobytes)
 - Allocated by SQL Server processes (in kilobytes)
- Disk Busy (percentage)
- VM CPU Usage (percentage)
- VM Memory Usage (percentage)

CPU Metrics

The CPU filter presents key metrics representing the most important performance indicators for the processors on this server.

- CPU Usage
 - SQL Server (percentage)
 - Total (percentage)
- Processor Queue Length (size)
- Processor Time
 - Privileged (percentage)
 - User (percentage)
- Call Rates
 - Compiles (number of, per second)
 - Re-Compiles (number of, per second)
 - Batches Received (number of, per second)

Disk Metrics

The Disk filter presents key metrics representing the most important performance indicators for the physical disk on this server.

- SQL Server Physical I/O
 - Page Reads (average number per second)
 - Page Writes (average number per second)
 - Checkpoint Writes (average number per second)
 - Lazy Writer Writes (average number per second)
 - Read Ahead Pages (average number per second)
- SQL Server Physical I/O errors (number of)
- Disk Reads Per Second (number of)
- Disk Transfers Per Second (number of)
- Disk Writes Per Second (number of)
- Disk Milliseconds Per Read (average time spent per disk read, in milliseconds)
- Disk Milliseconds Per Transfer (average time spent per transfer operation, in milliseconds)
- Disk Milliseconds Per Wait (average time spent waiting for process to finish, in milliseconds)

Memory Metrics

The Memory filter presents key metrics representing the most important performance indicators for your server memory.

- Memory Usage
 - Used by SQL Server (in kilobytes)
 - Allocated by SQL Server processes (in kilobytes)
 - Total Used (percentage)
- Paging (number of paging operations per second)
- Buffer Cache Hit Ratio (percentage)
- Procedure Cache Hit Ratio (percentage)
- Page Life Expectancy (in seconds).


Session Metrics

The Session filter presents key metrics representing the most important performance indicators for sessions connected to this server.

- Response Time (in milliseconds)
- Active Sessions (number of)
- Blocked Sessions
 - Block Sessions (number of)
 - Lead Blockers (number of)
 - Total Deadlocks (number of)
- Packets Sent (number of)
- Packets Received (number of)
- Batches Received (number of)

My Servers

The My Servers view shows a summary of all the monitored SQL Servers you monitor and their current health. Critical servers are marked with a red X while servers in good health are marked with green checks. This helps you to identify which instances are in critical state and focus your attention on the servers that they need immediate attention.

 Use the filter control to group your servers by severity or by the servers you are following.

You load this view by default when you log on to SQLDM Mobile using either a mobile device or your smartphone. Also, you can access this view from the SQLDM Mobile home page by tapping **My Servers**. Then, tap an individual server to access its dashboard.

Server dashboard charts


The Charts version of the Dashboard tab on the Server view is modeled after the Server Overview in the SQLdm Console. This duplication allows you to track the same performance metrics from the same charts whether you are at your desktop or away from your environment. Consistent monitoring is key to understanding the true performance and status of your SQL Server instances.

These charts are grouped, per performance category, into the following dashboards:

- Cache
- CPU

- Databases
- Disk
- Lock Waits
- Memory
- Network
- Server Waits
- Sessions

To get more information about the metrics in a specific chart, tap the chart to drill down to the associated, detailed views. On the server profile view, tap the **Dashboard** tap. The charts version loads by default on tablet devices and on your desktop browser. To get to the traditional Server Dashboard tap the **status** link located next to the server status.

 The chart server view is not available on small form devices such as smart phones.

Server dashboard status

The Status version of the Dashboard tab on the Server view provides a quick synopsis of the health of the server you are monitoring. From this view, you can:

- See the overall server status (as of the last refresh date and time).
- View a summary of active alerts.
- Choose to see specific alerts by severity.
- Get health status of key performance metric areas, such as sessions and memory.
- Choose to see specific performance metrics by area.
- Review the server properties, such as the host operating system, how many databases are managed by this server, the corresponding virtual machine properties (when applicable), and whether OS metrics are being collected by SQLDM for this server.

On the server profile view, tap the **Dashboard** tab, the status version loads by default on smar phone devices. On other devices, such as tablets and desktop browser tap the **Status** link on the Dashboard tab.

You can check the current alert status without waiting until the next shedule refresh. In the action menu, tap the **Refresh Alerts** option.

SQLDM Dashboard


The SQLDM Dashboard view in SQLDM Mobile displays the overall health of the servers monitored by your SQLDM application. You can assess server health by its status or by alphabetical order.

Each server summary display the three most recent alerts. Simply, tap the server summary to open the server dashboard view. To get to the SQLDM dashboard from your directory, tap the **SQL Diagnostic Manager** icon.

Sessions

The Session view displays the following information about the session you selected:

- The user and application that initiated the session.
- The start time of the session.
- The date and time of the last activity in the session.
- The last SQL statement executed by this session.

 In this view, you can kill a user session to prevent ongoing or escalating performance issues.

Sessions by type

The sessions view displays sessions per the selected type and provides a summary of their current status. Each session summary includes the following metrics:


- Session ID
- Name of the database user whose credentials are being used to execute statements or operations on this server
- Name of the database being used by this session
- Name of the application or job that initiated this session
- Total amount of I/O – disk reads and writes — initiated by actions taken during this session
- Total amount of CPU time required to process actions taken during this session
- Total amount of memory that is allocated to this session

The following are the type of sessions you can view:

- All active sessions
- All blocked sessions
- All blocking sessions
- Any session that is caught in a blocking chain
- Top 10 sessions consuming the most CPU
- Top 10 sessions consuming the most I/O
- Top 10 sessions consuming the most memory
- Top 10 sessions that are waiting for a resource

Also, you can access to the black chain view. This view displays a hierarchical list of the order in which session are blocked. The list starts with the lead blocker and expands the subsequently blocked sessions. In this way, you can see the relationships between blocking and blocked sessions and determine how extensive is the block.

Tap the blocked or blocking session you want, to view more details.

 In SQLDM Mobile, session performance metrics, including the number of sessions, are collected and displayed in real-time as you navigate from view to view. The connection lifespan of a session is typically brief, that may cause discrepancy between sessions.

Sessions Dashboard

The Dashboard tab on the Sessions view presents the most important performance indicator for sessions.

You can view the current value of these metrics from now up to 24 hours in the past. This allows you to identify performance patterns or unexpected peaks in activity.

- 30 minute period, with a data point every 5 minutes
- 1 hour period, with a data point every 10 minutes
- 12 hour period, with a data point at each refresh interval (by default, every 6 minutes)
- 24 hour period, with a data point at each refresh interval (by default, every 6 minutes)

Metrics available from this dashboard tab include:

- Response Time (in milliseconds)
- Clients (number of clients connected)
- Sessions
 - Active Sessions (number of)
 - Idle Sessions (number of)
 - System Sessions (number of)
- Block sessions

- Blocked Sessions (number of)
- Lead Blockers (number of sessions that are the lead, or head, of a blocking chain)
- Total Deadlocks (number of)

Sessions details

The Details tab on the Sessions view provides a launch-pad-like interface that lets view more details about a session.

- **Active User Sessions** - Lists all active sessions initiated by database users on this server, providing a performance summary for each session and a link to detailed session properties. From that view, you can also kill a user session.
- **Blocked** - Lists all blocked sessions on this server, providing a performance summary for each session and a link to detailed session properties. From that view, you can also kill the blocked session.
- **Blocking** - Lists all blocking sessions on this server, providing a performance summary for each session and a link to detailed session properties. From that view, you can also kill the blocking session.
- **Blocking Chain** - Displays the hierarchical relationship between the blocking sessions and the blocked sessions, from the session that started this blocking chain to the most recently blocked session affected by it.
- **Top CPU Consumers** - Lists the 10 sessions that are consuming the most processing resources on this server, providing a performance summary for each session and a link to detailed session properties. From that view, you can also kill the session.
- **Top I/O Consumers** - Lists the 10 sessions that are creating the highest I/O load on this server, providing a performance summary for each session and a link to detailed session properties. From that view, you can also kill the session.
- **Top Memory Consumers** - Lists the 10 sessions that have allocated the most SQL Server memory on this server, providing a performance summary for each session and a link to detailed session properties. From that view, you can also kill the session.
- **Top Waiting Sessions** - Lists the 10 sessions that have been waiting the longest time (in seconds) for a resource to become available to execute their statements or commands.

SQL Server Agent

The SQL Server Agent tab on the Services view displays the overall status of the SQL Server Agent as well as a status summary for each job. Swipe down the list to quickly see which jobs are currently running, which are scheduled to run later, and which have failed.

You can filter the list to show only those jobs that are running or have failed, or search for a specific job. You can also stop or restart the SQL Server Agent from this tab.

The Agent Job view provides detailed information about the job properties, the status of its last run, and its performance history. You can also stop or restart the selected job from this view.

Tap the job you want to see detailed Agent Job information.

Customize your experience

You can customize your IDERA Newsfeed experience to match your management style and monitoring needs.

The following features are available for customization:

- [Add servers and coworkers to the SQLDM Mobile home page.](#)
- [Follow the people and the servers you want.](#)
- [Use notifications to keep you up-to-date about specific server or coworker activity .](#)

Add servers and coworkers to the SQLDM Mobile Home Page

By adding servers and coworkers to your favorites list, you create shortcuts from the SQLDM Mobile home page to their walls. These shortcuts give you quick and easy access to recent status updates and posts from your mobile device when you are out of the office or stuck in a meeting. With SQLDM Mobile, no matter where you are, you can stay up-to-date about mission-critical SQL Server instances throughout your environment.

To add favorites using the IDERA Newsfeed:

1. From the Newsfeed navigation pane of the SQLDM Console, click **Edit Favorites**.
2. For each server or coworker you want to add, click **Favorites**.

To add favorites using SQLDM Mobile:

1. On the SQLDM Mobile home page, tap **Directory**.
2. Filter your list to show either Coworkers or SQL Servers.
3. Tap the name of the person or thing you want to add as a favorite.
4. On the action menu of the Wall view, click **Add Favorite**.

Change who and what you follow

Changing who and what you follow can completely change your IDERA Newsfeed experience. When you follow specific servers or coworkers, the IDERA Newsfeed Platform publishes their status updates and posts in your Most Recent feed. By selecting the instances you need to monitor and those coworkers with whom you frequently interact, you can ensure your feed contains the most relevant information you need to perform daily tasks and catch unexpected issues before they become crises.

How do I follow someone or something?

In the Newsfeed navigation pane of the SQLDM Console, click either **Coworkers** or **Applications** or **SQL Servers** , and then click **Follow** next to the name of the coworker, application, or server you want to follow.

How do I stop following someone or something?

In the Newsfeed navigation pane of the SQLDM Console, click either **Coworkers** or **Applications** or **SQL Servers** , and then click **Unfollow** next to the name of the person or thing you no longer want to follow.

Change your Notification Settings

When configuring your notification settings, identify the type of activity about which you want to receive notifications, and then enable the corresponding notification settings.

Ensure you get notified when specific activity occurs

Your Goal	Your Activity	Notification Settings You Should Enable
Track all activity on a critical issue	Bookmark the related story	Comments on a story you have bookmarked
Get a play-by-play account of how your team is resolving an active issue	Comment on the related story	Comments after you comment on a story
Learn about activity on a server you own	Own a server	Comments on a post on the wall of something you own Mentions something you own in a post or comment Posts a status update for something you own
Get an answer to your question about a specific issue	Comment on the related story	Comments after you comment on a story
Know when a coworker's contact information or profile name changes	Follow the coworker	Profile changes for people you follow
Confirm that your coworkers agree with your suggestion	Comment on the related story	Likes your post or comment

Ensure someone else gets notified

Your Goal	Your Activity	Notification Settings Your Coworker Should Enable
Start a conversation	Post to your coworker's wall.	Posts on your wall.
Assign an issue to a team member	Mention your coworker in a comment you add to a story.	Mentions you in a post or comment.
Ask your coworker a question about a server	Either mention your coworker in a post you add to the server's wall, or mention the server in a post you add to your coworker's wall.	Posts on the wall of something you own. Mentions you in a post or comment. Mentions something you own in a post or comment.

How do I change my notification settings?

To change which notifications you receive, click **Edit Settings** in the My Notifications view in the SQLDM Console.

How do I disable notification emails?

Deselect the Enabled checkbox for the Send notifications to setting on the My Notifications view in the SQLDM Console. When this setting is disabled, the SQLDM Mobile & Newsfeed Service will no longer send you notification emails about activity for the people and things you follow. You can continue to review your most recent notifications using the My Notifications view.

Tips and tricks

There are several ways you can use SQLDM Mobile with IDERA Newsfeed to make your life as a SQL Server DBA easier. The following tips and tricks can interest you:

- [Announce maintenance plans](#)
- [Ignore specific stories](#)
- [Search for the stories you want](#)
- [Inform your coworkers about your server status](#)
- [Use bookmarks to track critical servers](#)
- [Kill a session remotely](#)
- [Run queries to fix issues on remote servers](#)
- [Start or stop a service remotely](#)

Announce maintenance plans

You can easily announce upcoming maintenance plans for a monitored SQL Server instance.

- **Announce a server maintenance** - you can post in a server wall, mentioning key coworkers who can be affected, in the wall of your team leader, mentioning the server, or you can post to your own wall, mentioning the server and key coworkers.
- **Announce an application maintenance**- You can post on the application wall or on the wall of your team leader, mentioning the application.

Who gets notified?

Your Action	Who Gets Notified
Post to the wall of the server, mentioning key coworkers.	The server owner and the coworkers you mentioned.
Post to the wall of an application.	The application owner.
Post to the wall of your team leader, mentioning the server.	The team leader and the server owner.
Post to the wall of you team leader, mentioning an application.	The application owner.
Post to your own wall, mentioning the server and key coworkers.	The server owner and the coworkers you mentioned.

Ignore specific stories

You can easily ignore specific stories by unfollowing them. Unfollowing a story removes the story from you Most Recent Feed. You can unfollow a story even though you are following the server or person who initiated the story.

How do I ignore specific stories?

Find the story in you Most Recent Feed and then click the X next to the story summary.

How do I follow a story later?

If the ignored story becomes important later and you want to follow it again, click **Follow** in the story summary on the application, server, or the wall of your coworker. Also, you can bookmark a story, so you can access it later.

Search for specific stories

To search a specific story, you can easily revisit performance issues and see how they were resolved. You can search for current and past stories even if they were originated by servers or coworkers you are not following. Also, you can use the activity timeline to find stories that occurred on specific dates, or use bookmarks to track stories.

How do I access the Search feature?

In the SQLDM Console, the Search feature is available at the top of each view of the IDERA Newsfeed interface.

Inform your coworkers about your server status

You can easily warn your coworkers about a critical server status by mention them in a comment on the related status update from the server, post to their walls, or post to the server wall and mention your coworker in the post.

Use bookmarks to track critical servers

Following the actions you can perform when bookmarking a specific story:

- Quickly access to a critical issue without having to scroll through your Most Recent Feed.
- Compile an informal to-do list of issues you want to track over the course of the day or period of time.
- Keep an ongoing, historical record of important issues and their solutions.

When a story is bookmarked, it is permanently saved so you can always return to it later. You can configure the IDERA newsfeed to send you notifications when activity occurs on a bookmark.

 The bookmarked stories are not groomed.

Kill a session remotely

You can use SQLDM Mobile to kill a session that is currently connected to a remote SQL Server database.

Following are the steps to kill a session:

1. Navigate to the **Dashboard** tab for the affected server.
2. Under Areas, tap **Sessions**.
3. On the Sessions view, tap the **Details** tab.
4. On the Details tab, click the type of session you want to kill.
5. On the Sessions view, swipe or search for the specific session, and then tap it.
6. On the Session view, click **Kill Session**.

Run queries to fix issues on remote servers

You can use SQLDM Mobile to run custom, ad-hoc queries on remote SQL Server instances. SQLDM Mobile also provides several default query templates that perform common administrative tasks on the target database.

The custom queries that you run frequently in your environment, you can save them as templates to rerun later. All query templates can be used by any SQLDM Mobile user. Creating query templates also lets your team easily use approved queries that have been tested and optimized for your environment.

Run a query

To run a query, specify the following parameters, and then tap **Run Query**.

- Select target SQL Server instance
- Select target database (SQLDM Mobile will run your query against the master system database, by default)
- Choose appropriate query template (optional)
- Limit result set to a specific number of rows

When can I run a query?

You can access the Run Query feature from the action menu on any server view in SQLDM Mobile, letting you immediately address a performance issue or resolve an alert.

Does SQLDM Mobile log the queries that are run against monitored servers?

Yes. SQLDM records every query you run in the application event log. Each log entry includes the following information:

- Version of the browser that sent the query
- Date and time when the query was run
- Whether the query belongs to a sequence of batches
- T-SQL code used in the query

Can I use the GO keyword in my query?

Yes. The GO keyword is supported by the Run Query feature.

What is a query template?

A query template is a T-SQL script saved to the SQLDM Mobile Repository. You can use the default query template to get started, and then later save your own query templates for customized troubleshooting and issue resolution.

SQLDM Mobile provides the following default template:

Template name	What it does?
Backup History	Checks when the most recent backup occurred on each database hosted by the selected SQL Server instance, and then calculates how many days have passed since that operation was run. Both values are returned, along with the name of the corresponding database.

How do I create a query template?

You can save your own query template that contains a T-SQL script you frequently use.

1. Type the appropriate T-SQL code into the query text box.
2. Tap the **Template** pull-down menu, and then tap the **Save As** template option.
3. Type a new name, and then tap **Save**.

Can I delete a template I no longer need?

Yes. Use the **Template** pull-down menu to select the template you want to delete, and then tap the **Delete** template option.

Start or stop a service remotely

You can use SQLDM Mobile to start or stop a SQL Server Agent service that is located on a remotely monitored server.

- ✓ When the selected service is running on a virtual SQL Server instance (A SQL Server instance hosted by a Windows Server cluster), you cannot stop or start the service using SQLDM. Instead, use the native Microsoft SQL Server tools to manage this service.

Follow these steps, to stop or start a service:

1. Navigate to the Dashboard tab for the affected server.
2. Under Areas, tap **Services**.
3. On the Services view, tap the **SQL Server Agent** tab.
4. Under Agent Status, move the toggle to either Stop or Start.

Monitor SQL Server Performance

SQL Diagnostic Manager allows you to analyze the performance of your monitored SQL Server instances from My Views and from the Dashboard view.

See the general status of all your monitored SQL Server instances

The SQL Diagnostic Manager Today view allows you to view the status of all your monitored SQL Server instances, including a listing of all your active alerts.

Get the configuration properties of all your monitored SQL Server instances

The Server Configuration Properties view allows you to view the important configuration settings for all of your monitored SQL Server instances. While you manage these settings in different areas of SQL Diagnostic Manager, you can see a current list of all of your instances and their properties settings in one view.

Get an overview of a group of SQL Server instances

My Views allows you to monitor either specific groups of SQL Server instances, or all groups. This overview includes monitors that allow you to view several important specifics, see at-a-glance if there are any problems, and drill down to detailed metrics for each component on the SQL Server instance.

Get an overview of a specific SQL Server instance

The Dashboard view provides a graphical representation of the activity on your SQL Server instance. This dashboard allows you to diagnose issues that are inter-related, using quick links to see the associated details and review possible solutions to specific issues.

Interact with the selected SQL Server instance using IDERA Newsfeed

On each server view, you can interact with the selected SQL Server instance by either:

- Viewing the server profile
- Posting to the server's wall
- Following the server


For more information, see the [IDERA Newsfeed Help](#).

Launch SQL Workload Analysis

SQL Diagnostic Manager allows you to launch SQL Workload Analysis in context of your monitored server from the desktop console and the web console. SQL Workload Analysis dashboard displays trending database activity and top utilizing SQL statements, logins, machines, and programs. SWA provides a granular breakdown of the SQL Server wait states with easy drill-down isolate problems quickly. View real-time and historical data to help tune queries and actionable recommendations to improve performance. For more information, see the [SQL Workload Analysis Online Help](#).

Launch SQL Workload Analysis

In the desktop console, right-click the All Servers option under MY VIEWS, and select **Open** to get the Thumbnails view of all servers.

SQL Diagnostic Manager recognizes which servers are being monitored by SQL Workload analysis as well. The  displays only under these servers thumbnails.



In the list view of the web console, the listed instances are well identified with the SWA icon when they are registered in SQL Workload Analysis.

SWA	Instance Name	Health	Alert	CPU	Mem	I/O	Dat...	Logs	Qu...	Version	Status	Agent Status	DTC Status	Availa
	WIN-513F1007B00	0.00%								14.0.1000.1...	Running	Running	Running	521912
	SIDROCTST02\SQLE2014	37.50%								12.0.2000.8	Running	UnableToM...	UnableToM...	0
	SIDROCTST03\SQLE2014	37.50%								12.0.2000.8	Running	UnableToM...	UnableToM...	0

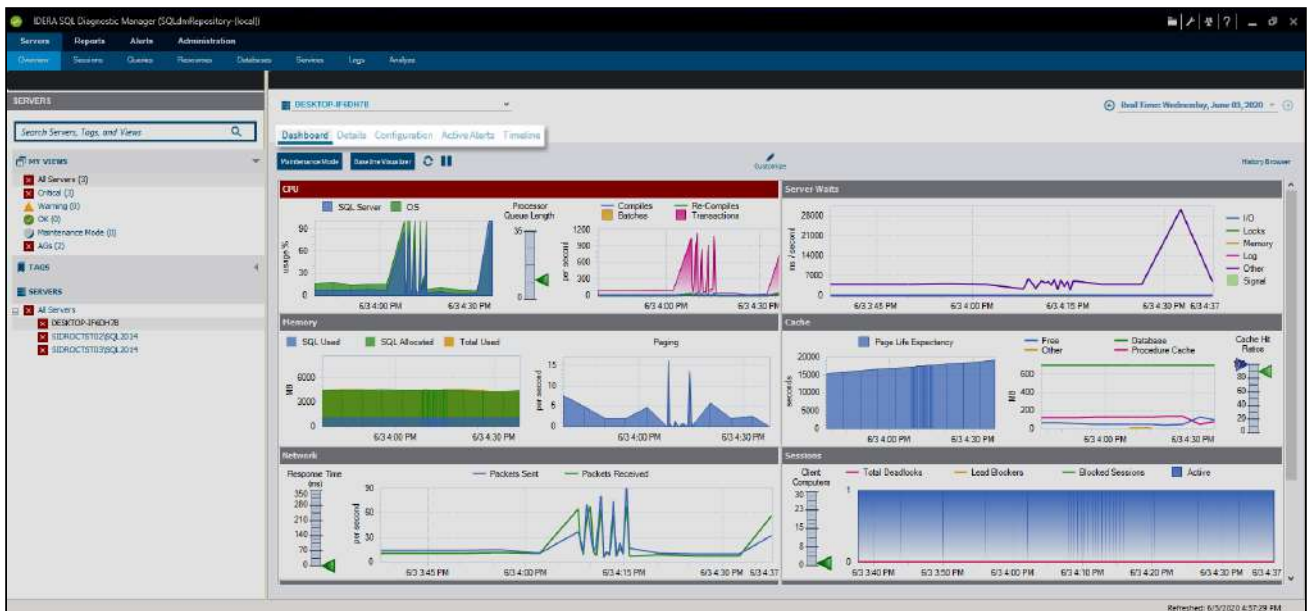
Monitor instances

The Overview tab contains the following views:

- Dashboard view where you can see an overview of the instance.
- Details view that lets you create charts with the specific data items you need.
- Configuration view that allows you to view the configuration information of your SQL Server.
- Active Alerts view where you can see all outstanding alerts for this instance.
- Timeline view where you can see a timeline of events as they occurred on this instance.

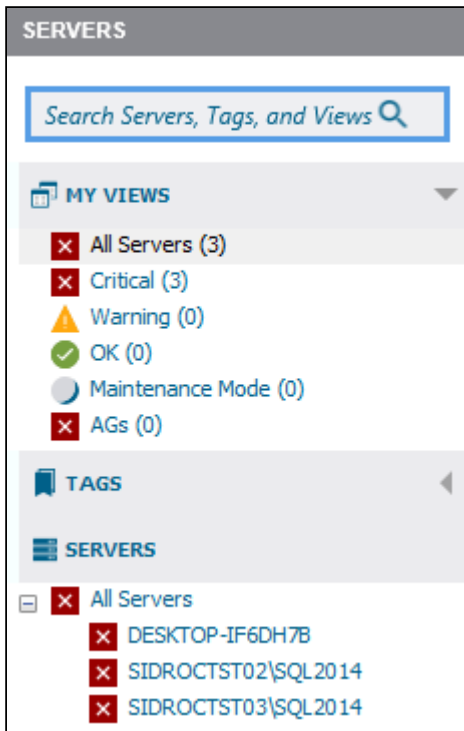
Access the Overview tab

Access the Overview tab by clicking the appropriate monitored SQL Server instance, this tab is the default view when you first click an instance.



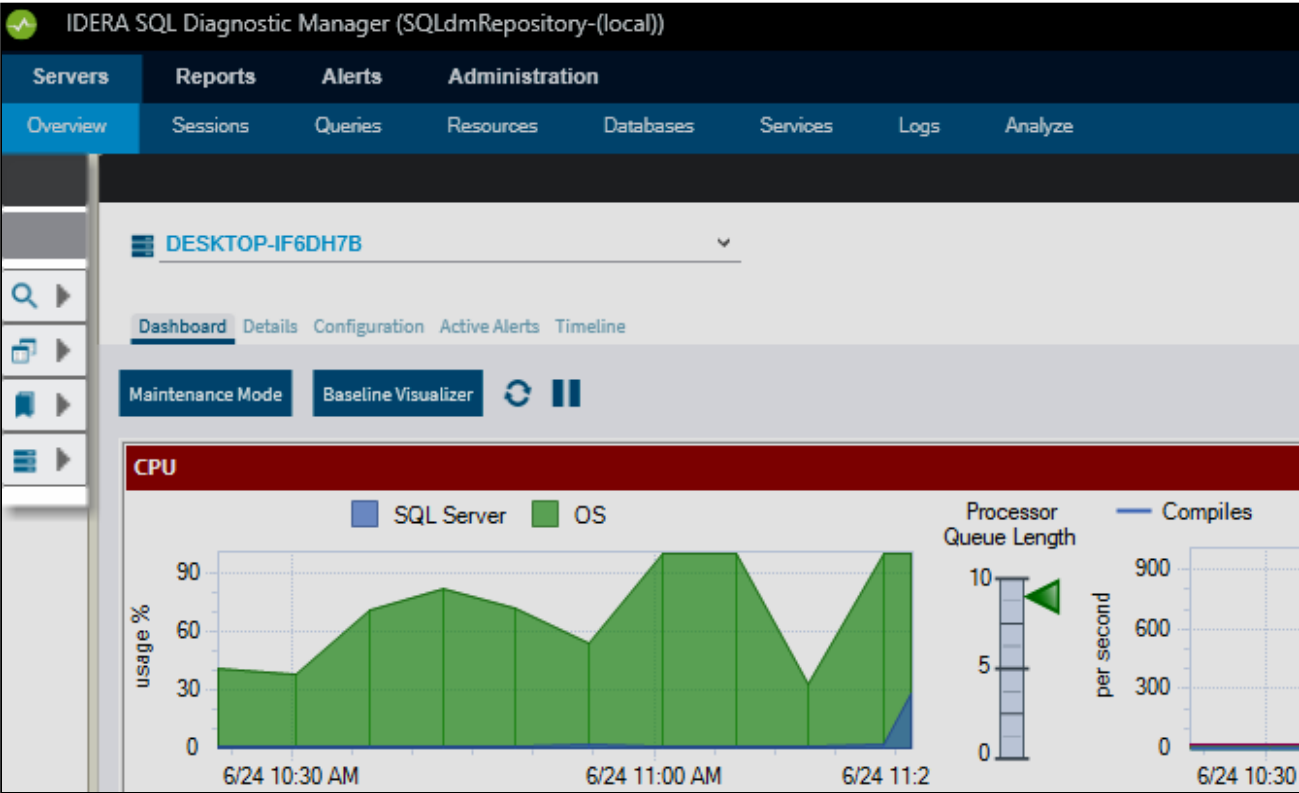
Navigate to your SQL Server instances

The SERVERS sidebar contains all the SQL Server instances that are monitored by SQL Diagnostic Manager, providing a general overview:

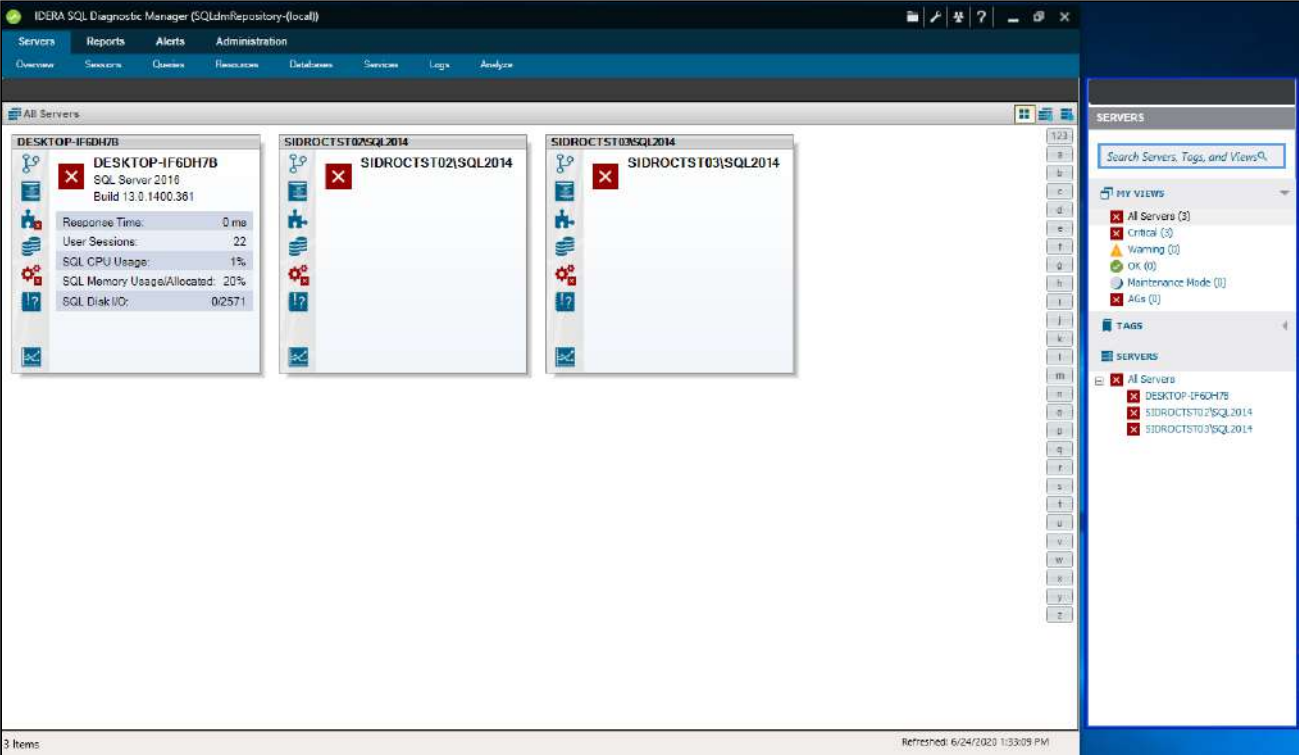


- The SERVERS panel includes a search bar that eases the search in environments with a large number of instances.
- Under MY VIEWS, select All Servers to see all your instances thumbnails with fast information, review more about the [available views](#). Additionally, MY VIEWS displays status icons, when a SQL Server instance is associated with a Critical, Warning, OK, and Maintenance Mode metric icon, you can use your mouse cursor to hover over the server name to view the top three associated alerts, allowing you to prioritize your SQL Server instances and the work that you need to perform.
- The Servers sidebar displays [tags](#), which are a way of labeling and organizing servers.
- Under Servers, you can find the list of the monitored servers by SQL Diagnostic Manager, click any of your servers name to access the Overview screen of the selected server.

The Servers sidebar can be expanded or collapsed, allowing you to have a larger screen space.



Additionally, SQL Diagnostic Manager allows you to undock and move your panels as you prefer. Double-click the dark gray bar to undock the panel, move it wherever you preferred, and double-click to return to its original location:



Refresh your alerts

A full refresh of alerts excluding table fragmentation occurs in the following circumstances even if longer collection intervals are defined:

- When the collection service starts
- When the collection service receives a new workload, such as redirecting the Management Service to a new Repository
- When you add a SQL Server for collection
- When a SQL Server resumes from Maintenance Mode
- When a user selects Refresh Alerts

This refresh does not cause collection of non-alertable data, nor cause collection of anything that is disabled.

Overview tab features

When you select a SQL Server instance in the Servers sidebar, the Overview tab displays the following options:

Dashboard

The [Dashboard view](#) is where you can see a graphical representation of the most important metrics for the selected SQL Server instance.

Details

The [Details view](#) lets you view the values of each data item collected and create charts with the specific data items you need.

Configuration

The [Configuration view](#) allows you to view and control the operational configuration of the monitored SQL Server instance.

Active Alerts

The [Active Alerts view](#) allows you to view and control the active alerts as of the most recent data collection.

Timeline

The [Timeline view](#) allows you to view all alerts and maintenance mode events as well as the first monitored event for the selected SQL Server instance.

Other Servers features

By right-clicking a SQL Server instance in the Servers sidebar you can perform the following actions:

Open

 opens the Dashboard view for the selected monitored SQL Server instance.

Refresh Alerts

Gathers the alert status and refreshes the tree view information for the selected monitored SQL Server instance.

Delete

Allows you to delete the SQL Server instance from your SQL Diagnostic Manager installation. When you select **Delete**, SQL Diagnostic Manager displays a message asking whether you want to retain the collected data for the SQL Server instance, and it also allows you to cancel out of the deletion process.

- ✔ You cannot retrieve collected data once it is deleted from the SQL Diagnostic Manager Repository. Use the Delete option with care.

Maintenance Mode

Allows you to take an individually monitored SQL Server instance offline for scheduled maintenance or other reasons.

Configure Baseline

Allows you to configure the performance baseline for the selected SQL Server instance.

Apply Alert Template

Allows you to select an existing alert template to apply to the selected SQL Server instance.

Configure Alerts

Allows you to specify the alert criteria for each collected metric on the selected SQL Server instance.

Snooze Alerts

Allows you to snooze all the alerts for the selected SQL Server instance instead of selecting the snooze feature for each alert individually.

Resume Alerts

Allows you to resume any alerts set to snooze on the selected SQL Server instance.

Properties

Allows you to view and edit the collection properties of your SQL Server instance.

Get the SQL Server performance overview

The Dashboard view contains a graphical representation of the activity on your SQL Server instance. This view not only displays the status of your SQL Server instance, but it allows you to diagnose where problems originate.

The Dashboard view is made up of panels, which include additional information for specific metrics in [chart form](#). Hover your mouse cursor over an area within a chart on which you want to get actual statistics. Click a specific metric to drill down for more detailed information. By using the options in the menu available when you right-click over an area, you can configure alerts, view associated help topics and details, print, save the image, or export the dashboard to Microsoft Excel as a .csv file.

You can also customize your Dashboard view to show only those panels that are important to your monitored SQL Server instance. SQL Diagnostic Manager also allows you to save your customized view on a per-instance basis or use the same view for all of your instances. For more information about customizing your Dashboard view, see [Customize your Dashboard view](#).

Access the Dashboard view

You can open the Dashboard view of the SQL Diagnostic Manager Overview tab by selecting the appropriate SQL Server instance, and then clicking **Overview > Dashboard**.

Dashboard view

The Dashboard view is organized with panels that display all the key metrics collected by SQL Diagnostic Manager by category.

The available data panels for the Dashboard view include:

Cache

The [Cache panel](#) tracks the database and procedure processes using a buffer pool of memory on your monitored SQL Server instance, and includes the following charts and gauges:

- Page Life Expectancy
- Areas
- Hit Ratios

CPU

The [CPU panel](#) includes key CPU metrics for your SQL Server instance and the computer that hosts the instance. The metrics displayed in CPU include the following charts and gauges:

- Usage
- Processor Queue Length
- Call Rates

Custom Counters

The [Custom Counters panel](#) tracks selected custom counters for your monitored SQL Server instance. You must have at least one [custom counter](#) to view data in this panel.

Databases

The [Databases panel](#) includes key metrics covering the health of the databases on your monitored SQL Server instance. The metrics displayed in Databases include the following charts and gauges:

- Transactions/sec

- Log Flushes/sec
- Reads/sec
- Writes/sec
- I/O Stall ms/sec

Disk

The [Disk panel](#) includes key metrics that monitor the health of the disk of your monitored SQL Server instance. The metrics displayed in Disk include the following charts and gauges:

- Latency
- Throughput
- SQL Server Physical I/O

File Activity

The [File Activity panel](#) includes the top five database files with the highest relative activity since the last refresh of your monitored SQL Server instance. You can select different metrics and display results for all files, or a single database, disk, or file. The available metrics include:

- Reads/sec
- Writes/sec
- Transfers/sec

Lock Waits

The [Lock Waits panel](#) tracks the total number waits caused when a task is waiting to acquire a lock on your monitored SQL Server.

Memory

The [Memory panel](#) includes key Memory metrics for your SQL Server instance and the computer that hosts the instance. The metrics displayed in Memory include the following charts:

- SQL Server Usage
- SQL Server Paging
- VM Memory Usage
- Host Memory Usage

Network

The [Network panel](#) tracks the performance of the network connection being used by your monitored SQL Server, and includes the following charts and gauge:

- SQL Server Throughput
- VM Network Usage Throughput
- Host Network Usage Throughput
- Response Time

Server Waits

The [Server Waits panel](#) tracks the total number and time spent on waits affecting your monitored SQL Server.

Sessions

The [Sessions panel](#) includes session-specific data items. The metrics displayed in Sessions include the following chart and gauge:

- Activity & Blocking
- Clients

Tempdb

The [Tempdb panel](#) tracks the status of the tempdb database on your monitored SQL Server instance, and includes the following charts and gauge:

- Tempdb Space Used
- Tempdb Contention
- Version Store Cleanup Rate

Virtualization

The [Virtualization panel](#) tracks the performance of the host server that manages your virtual machines and the VM on which the monitored SQL Server instance is running, and includes the following charts and gauge:

- Virtual Machine Memory Areas
- VM Host Memory Areas
- Virtual Machine Disk Usage
- VM Host Disk Usage
- CPU Ready

Understand the Dashboard colors

The following colors are associated with a status for each of the provided panels.

Gray

Indicates the status of the data is normal, or within the thresholds assigned to the metric.

Blue

Indicates the status of the data is at an informational level based on your settings.

Yellow

Indicates the status of the data is at a warning level, or outside the threshold of normal but not yet at the critical status.

Red

Indicates the status of the data is at a critical level and requires immediate action.

Blank metrics on the Dashboard

If a metric on the Dashboard has no value, or displays as blank, SQL Diagnostic Manager is not able to collect the corresponding counter. Causes of this situation include:

- The SQL Diagnostic Manager Console has not captured the metrics. It can take up to two refreshes for this capture to occur.
- OS metrics are disabled. SQL Diagnostic Manager requires direct WMI or OLE automation to collect data.
- If using OLE automation, those procedures are unavailable. SQL Diagnostic Manager is unable to locate required OLE automation stored procedures.
- If connecting to WMI directly, the WMI Service is not running. SQL Diagnostic Manager requires that the WMI service is running on the computer that hosts the SQL Server instance in order to collect OS metrics.

- The counters do not exist on this SQL Server instance.

You can [enable OS metrics monitoring](#) to track the performance of the computers on which your monitored SQL Server instances are hosted.

Refresh your Dashboard data

The Dashboard view is refreshed according to the time set in the **Server View Refresh** field on the Console Options window, which is accessible by clicking **Tools > Console Options**.

Understanding your chart data and refresh interval

SQL Diagnostic Manager charts provide data whether collected real-time or as part of a scheduled collection event. For additional information about how SQL Diagnostic Manager displays your chart data, see [Charts](#).

Customize your Dashboard view

The Dashboard view is customizable per monitored SQL Server instance by selecting the panels that are important to view for that instance. You can customize your Dashboard view for the selected SQL Server instance or as the default for all SQL Server instances added to SQL Diagnostic Manager.

Using the Panel Gallery

When you install SQL Diagnostic Manager, a default Dashboard view is applied to all your SQL Server instances. The Panel Gallery allows you to alter your view so that you can monitor the most important metrics. To access the Panel Gallery, select your preferred SQL server instance from the Servers pane and then click **Customize**. Once in the Dashboard Designer, click a panel in the Panel Gallery for SQL Diagnostic Manager to display additional information about that panel.

Designing a Dashboard view

When you design a Dashboard view, select the metric panels that display the current status and the best information for your selected SQL Server instance. SQL Diagnostic Manager allows you to create multiple designs, you can create one design with panels specifically for SQL Server 2000 instances and another design for SQL Server 2005 instances. It allows you to create designs that show the most important panels to monitor when certain events trigger alerts.

SQL Diagnostic Manager allows you to choose from several different layouts of columns and rows for your dashboard design. You can then drag and drop panels from the Panel Gallery to the actual location in the layout where you want that panel to reside. You can also click, hold, and drag panels to re-arrange their location. When you drag a panel to another position and release the mouse button, the panel you moved switches locations with the existing panel. For example, your design includes the CPU panel in the upper left corner and the Sessions panel in the lower right corner. Click, hold, and drag the Sessions panel to the upper left corner, and then release. Note that the Sessions panel is in the upper left corner and the CPU panel is now in the lower right corner. For more information on available designs, see [Design a dashboard view](#).

SQL Diagnostic Manager includes several options for you to use as your Dashboard view. You can also create a design using the available panels, and save the design for other people in your environment to use as their Dashboard view. For more information on available designs, see [Select a dashboard design](#).

Saving a design

Once you are pleased with the selection and positioning of your panels, you can save your design. In the Dashboard Designer, click **Save**. SQL Diagnostic Manager allows you to name this design, which is then displayed in the Dashboard Gallery. For additional information about saving a dashboard design, see [Save a dashboard design](#).

Understanding how you can apply a dashboard design

SQL Diagnostic Manager allows you to use a design for a single session of the SQL Diagnostic Manager Management Console, apply a design to a specific monitored SQL Server instance, or apply the layout to any other SQL Server instance including those you add later. You can use a custom Dashboard view in the following situations:

One-time use

SQL Diagnostic Manager allows you to view a custom layout of your Dashboard view for a single session of the SQL Diagnostic Manager Management Console. For example, an event in one of your instances triggered an alert. You want to watch the metric that includes statistics relating to this alert, so you access the Panel Gallery in the Dashboard Designer. Drag the panel displaying the alert-related statistics to the top-left corner of your layout, and then close the Dashboard Designer. SQL Diagnostic Manager

displays a message that you made changes to the current design. Click **No** to use this design only until you close this session of the SQL Diagnostic Manager Management Console.

SQL Diagnostic Manager keeps that panel in the position of your Dashboard view until you close the SQL Diagnostic Manager Management Console session. Even after accessing other areas within SQL Diagnostic Manager, when you return to your Dashboard view, the same panel appears in the same location.

When you no longer want that panel foremost in your Dashboard view, simply close the SQL Diagnostic Manager Management Console knowing that the default Dashboard view for that instance appears the next time you access SQL Diagnostic Manager.

Set as the default design for one or more instances

While the default Dashboard view includes panels that are important to your monitored SQL Server instances, the panels may not be in the order in which you want or this design does not include all of the panels you want to view. If there are no existing designs you want to use, access the Panel Gallery to select and move the panels you want to the positions that most suit your needs.

When you are done with your design, click **Save** to make sure you can use this design again. Type a unique name for this design, verify that this SQL Server instance appears in the **Selected Servers** area. You can add any other SQL Server instances, select the name of the instance, and then click **Add >** to move the instance to the Selected Servers area. Once your list of selected instances is complete, click **OK**.

Each time you access the Dashboard view for this instance, SQL Diagnostic Manager displays the information using this design.

You can also select an existing dashboard design as the default for one or more monitored SQL Server instances. On the Server Overview tab, click **Customize** to open the Dashboard Designer, and click **Select a Dashboard**. Scroll through the available designs and select the one you want to use for the monitored instance. Click **Select** and each time you access the Dashboard view for this instance, SQL Diagnostic Manager displays the information in this design.

Set as the default design for all new or unassigned instances

SQL Diagnostic Manager allows you to select a design to use as the default for all new or unassigned monitored SQL Server instances so that you have a consistent format across all of your monitored SQL Server instances. When saving a dashboard design, check the **Use this dashboard as my default for new or unassigned servers** check box.

Available panels

- [Cache panel](#)
- [CPU panel](#)
- [Custom Counters Panel](#)
- [Databases panel](#)
- [Disk panel](#)
- [File Activity panel](#)
- [Lock Waits panel](#)
- [Memory panel](#)
- [Network panel](#)
- [Server Waits panel](#)
- [Sessions panel](#)
- [Tempdb panel](#)
- [Virtualization panel](#)

Design a Dashboard view

Throughout the day, your monitored SQL Server instances can go through a number of situations including triggered alerts requiring immediate attention. Some events result in multiple alerts, each needing you to keep a close eye on as a resolution is put into place. To help you monitor your SQL Server instances, SQL Diagnostic Manager provides a default Dashboard view that includes commonly-monitored metrics for the selected SQL Server instance.

You can improve the Dashboard view by customizing your view to include those panels most important to you. A customized Dashboard view can help you keep your monitored SQL Server instances from triggering alerts by providing you real-time coverage of crucial metrics.

You can view your current Dashboard view by making the appropriate changes, and then closing the Panel Gallery pane for a full view of your design. Click **View Panel Gallery** to toggle the display of the Panel Gallery selection pane. You can also close the Panel Gallery by clicking the close icon in the upper right corner of the pane.

Identify which Dashboard panels you want to include

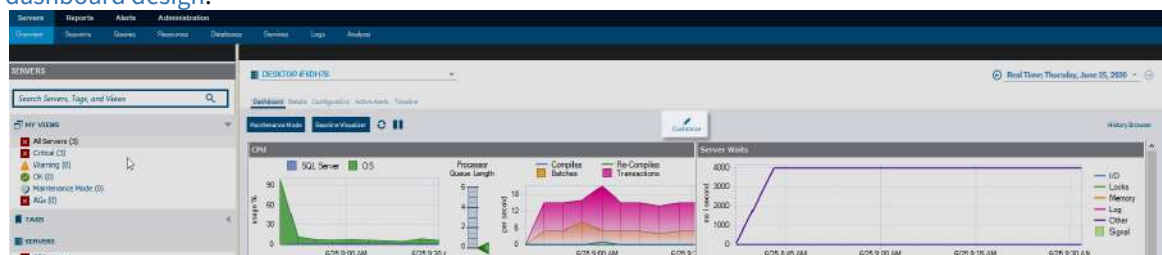
In the Panel Gallery, each panel includes a brief description and a link to access additional information about that panel. To view the description, click the panel for which you want to view details. SQL Diagnostic Manager also displays a sample graphic showing the kinds of charts and data available in the panel. Click **More info** in the description to access the help topic specifically for that panel.

Note that you can use the same panel more than once in your Dashboard design. Because some panels in the Panel Gallery provide drop-down selections for you to select metrics or other filters, you can use the same panel more than once in your Dashboard. This feature is handy when using panels such as the Databases panel because it allows you to have one Databases panel displaying a selected metric and another Databases panel displaying a different metric. You can do the same with the Disk panel to view the same metrics for different disks at the same time.

Manage your Dashboard view

To change your current Dashboard view:

1. Select the appropriate SQL Server instance from the Servers sidebar.
2. Click **Customize**, notice that you are now in the Customize Dashboard screen.
3. If you want to change the number of panels to display in your view, click **Select Layout**, and select the number of columns and rows.
4. Using a drag-and-drop method, move the panels you want to include from the Panel Gallery to the location within the view where you want that panel to reside. You can place the same panel more than once in your view. This feature is important on panels that allow you to select different metrics, such as the Databases panel which offers different measurements per second.
5. Click **Close Designer** once you select and move panels to where you want them to reside.
6. On the confirmation message, click **Yes** to use this design again or apply it to additional SQL Server instances, or click **No** to use this design for only this session of the SQL Diagnostic Manager Management Console. For additional information about saving a dashboard design, see [Save a dashboard design](#).



Select an existing dashboard design

SQL Diagnostic Manager includes a default Dashboard view and multiple designs with installation. If you want to change your view, you can select each panel specifically for a custom view or you can select from existing layouts that you or another user in your environment created. This feature allows you to share designs as templates that each user can use to save, and then apply to their monitored SQL Server instances.

Select an existing design for your Dashboard view

To use an existing design for your Dashboard view:

1. Select the appropriate SQL Server instance from the Servers pane.
2. Click **Customize**. Notice that you are now in the Dashboard tab. SQL Diagnostic Manager displays this tab when you use the Customize feature.
3. Click **Select a Dashboard**.
4. Scroll through the displayed designs until you find the one you want to use. Click the arrows to scroll through the available designs. Click **All Dashboards** to view designs other users in your environment created and made available for use. This feature allows you to "share" a design that you find effective. Click **Select** when you find the design you want to use. SQL Diagnostic Manager returns you to the Dashboard Designer using your selected design. You can then save this design to use for this SQL Server instance or click **Close Designer** to use this design only during this session of the SQL Diagnostic Manager Management Console. For additional information about saving a dashboard design, see [Save a dashboard design](#).

Delete a Dashboard layout

If you created a layout that you no longer want anyone to use, you can delete that layout from your gallery. Simply click **Select a Dashboard**, scroll to the layout you want to delete, and then click **Delete**. You cannot delete any of the layouts supplied with your SQL Diagnostic Manager installation.

Save a dashboard design

There are two options when saving a design for your Dashboard view. Check **Use this dashboard as my default for new or unassigned servers** if you want SQL Diagnostic Manager to use this design for all new SQL Server instances and for existing instances that do not have a designated default Dashboard view.

To save a dashboard design:

1. In the Dashboard Designer, click **Save**.
2. Type the name you want for your new design. If you use the name of the existing design, SQL Diagnostic Manager displays a message alerting you to the existing design. You can then type a different name or use the same name, replacing the existing design with this new format.
3. Verify that the correct monitored SQL Server instance appears in the Selected Servers area. You can apply this design to additional instances by selecting the instance, and then clicking **Add >** to move the instance from **Available Servers** to **Selected Servers**.
4. Check **Use this dashboard as my default for new or unassigned servers**, if you want SQL Diagnostic Manager to use this design as the default Dashboard view for all of your new monitored SQL Server instances and any instance that does not currently have a designated default design.
5. Click **OK**.

Cache panel

The Cache panel tracks the database and procedure processes using a buffer pool of memory on your monitored SQL Server instance. Unexpected spikes in the buffer and procedure cache may alert you to check that you have enough memory for the current pool to function properly. Chronically high cache metrics may indicate the need for server maintenance, query tuning, or index updates to better handle the ongoing workload. For additional information, see [Manage your procedure cache](#).

If the database cache size seems to be too small for optimal performance and there is very little available memory on the system, adding more memory to the system may increase performance. If there is a lot of available memory on the system and the database cache size is not growing beyond a certain point, the database cache size may be capped at an artificially low limit. Increasing this limit may increase performance.

Page Life Expectancy chart

The **Page Life Expectancy** chart provides the time in seconds that a page remains in the cache or buffer pool. The longer your page stays in the cache, the more likely the chance that SQL Server finds that data quickly without a read of physical media and minimize processing resource use. Use the [Page Life Expectancy alert](#) to warn you of issues.

Areas chart

The Areas chart displays the usage in megabytes of each type of cache in use.

Metric	Why it is important
Database	Indicates the amount of system memory (in MB) used by the database cache manager to hold commonly-used information from the database file(s) to prevent file operations.
Free	A low amount of free cache indicates a higher risk of performance issues.
Procedure Cache	Indicates the amount of space used to store query plans.
Other	Indicates the amount of space used by other processes.

Cache Hit Ratios gauge

The **Cache Hit Ratios** gauge displays the hit rates of the Buffer and Procedure cache in this SQL Server instance. Use the [Procedure Cache Hit Ratio \(Percent\) alert](#) to warn you of issues.

Available alerts

- [Page Life Expectancy alert](#)
- [Procedure Cache Hit Ratio \(Percent\) alert](#)

CPU panel

The CPU panel tracks the performance of the processor used by your monitored SQL Server. Unexpected spikes in CPU usage and call rates may warn you about the beginning of a serious performance issue. Chronically high CPU metrics may indicate the need for server maintenance, query tuning, or index updates to better handle the ongoing workload. For additional information, see [how to get your CPU performance details](#).

Usage chart

The **CPU Usage** chart displays the percentage of processing power in use on the computer that hosts the SQL Server instance over a period of time. The CPU view on the Resources view allows you to track your CPU usage over a period of time, along with other key CPU statistics. For additional information about how SQL Diagnostic Manager works with virtual machines and collects metrics, see [How SQL Diagnostic Manager works with a virtual environment](#).

Metric	Why it is important
SQL usage	A higher SQL usage can indicate that SQL Server is spending too much time processing queries.
Total usage	A high total CPU usage can indicate that this server does not have enough resources to adequately process its current workload.
VM usage	A high virtual machine usage can indicate that the processor power allocated to this VM is insufficient for its current workload.
Host usage	A high host server usage indicates the processing power on this host is insufficient to handle the workload of the currently active VMs.

Processor Queue Length gauge

The **CPU Processor Queue Length** gauge displays the current value of the [processor queue length metric](#). Use this metric to determine how much work is waiting on this server. A high processor queue could indicate a blocking session or other performance issues.

Call Rates chart

The **CPU Call Rates** chart breaks down the processor workload into the number of batches, compilations, and transactions completed each second, giving you a detailed view of which activities are resource-intensive.

Metric	Why it is important
Batches/sec	A high throughput rate can indicate a higher risk of network, CPU, and resource issues as the SQL Server performance degrades.
Compiles/sec	A high number of compilations (greater than 100 per second) can indicate a high server workload or that there is more recompilation overhead than necessary.

Metric	Why it is important
Re-Compiles/sec	A high re-compilation rate can indicate excessive re-compilation overhead.
Transactions/sec	A high rate of transactions can indicate a higher risk of resource issues, such as blocks or locks, due to the heavier workload.

Differences in statistics

You may notice some difference in your statistics, such as your OS CPU usage being higher than your VM CPU usage. While the hypervisor manages resources efficiently, the demand for physical resources may be greater than what it has to provide. For example, a guest OS on a virtual machine submits a batch of work to a CPU for execution. However, all of the physical CPUs are committed to other work, so there is no physical CPU available on the host to process the work. In this situation, the VM has to wait for a CPU to become available to process its work. While it is waiting, the guest OS is unaware of the wait and assumes that it is simply taking longer for the CPU to process this batch of work, and the OS thinks it is using more CPU power than it is actually using.

Remember that SQL Diagnostic Manager uses the additional overhead of the hypervisor to calculate the VM metrics. In this situation, the hypervisor knows the VM is just waiting for a CPU, so it does not charge the VM for CPU processing power while it is waiting. The result is that the guest OS reports that it is using the additional processing power.

Available alerts

- [Host CPU Usage \(Percent\) Alert](#)
- [OS Processor Time \(Percent\) Alert](#)
- [OS Processor Queue Length \(Count\) Alert](#)
- [SQL Server CPU Usage \(Percent\) Alert](#)
- [VM CPU Usage \(Percent\) Alert](#)

Custom Counters panel

The Custom Counters panel displays the statistics for a selected custom counter on a monitored SQL Server instance. At least one custom counter must exist on the selected instance for this panel to display any data. Custom counters allow you to include more performance metrics than are provided by SQL Diagnostic Manager. For additional information about custom counters, see [Use custom counters to track metrics](#).

Any alerts affecting the Custom Counters panel depend on the selected custom counter and alert threshold settings.

Databases panel

The Databases panel tracks the performance of the databases on your monitored SQL Server. Unexpected spikes in database usage may warn you about the beginning of a serious performance issue. This feature offers a number of different metric sets based on your preference, which allows you to view your database transactions, log flushes, reads, writes, or I/O stalls. For each metric set, you can view a list of your top 100 databases sorted by current usage of that metric, and a bar representing the relative size of the associated database. A trends chart displays the same data over a period of time for only your top five databases. For additional information, see [Get the database status summary](#).

Transactions chart (per second)

The **Transactions/sec** chart displays the number of transactions per second occurring on the databases on your SQL Server instance.

Log flushes chart (per second)

The **Log Flushes/sec** chart displays the number of log flushes performed per second and usually matches the number of transactions per second. If you experience a disk bottleneck, review to see if your log flushes greatly outnumber your transactions.

Reads chart (per second)

The **Reads/sec** chart displays the number of reads per second performed by the databases on your SQL Server instance.

Writes chart (per second)

The **Writes/sec** chart displays the number of writes per second performed by the databases on your SQL Server instance.

I/O stall chart (ms/sec)

The **I/O Stall ms/sec** chart displays the total time per second (in milliseconds) that the databases on your SQL Server instance were waiting on an I/O to complete. Chronically high I/O stall metrics may indicate a disk bottleneck.

Disk panel

The Disk panel tracks the performance of the disk used by your monitored SQL Server. Unexpected spikes in latency, throughput, or SQL Server IO may warn you about the beginning of a serious performance issue. Chronically high disk metrics can signify an insufficient IO subsystem, or excessive or inefficient IO activity on the server. Use the **Disk** view on the **Resources** tab to track several key disk statistics over a period of time. For additional information, see [Get the disk performance details](#).

Latency chart

The **Disk Latency** chart plots the average time (in milliseconds) required to complete disk reads and writes, tracking the disk IO speed over time.

Metric	Why it is important
Average Disk ms/Read	Higher disk read times impact the ability of SQL queries to retrieve data from your databases.
Average Disk ms/Write	Higher disk write times impact the ability of SQL queries to update your databases, from simple text column modifications to dropping tables.

Throughput chart

The **Disk Throughput** chart plots the number of disk reads and writes over time.

Metric	Why it is important
Disk reads/sec	A high number of disk reads may indicate that the SQL Server is executing poorly-performing queries.
Disk writes/sec	A high number of disk writes may indicate that the SQL Server is running low on allocated memory .

SQL Server Physical I/O chart

The **SQL Server Physical I/O** chart breaks down the SQL Server throughput into its key component pieces, giving you a detailed view of how well SQL Server is processing your data requests and updates.

Metric	Why it is important
Checkpoint writes	A high number of checkpoint writes may indicate that pages are not written to disk quickly enough for optimal performance.
Lazy writer writes	A high number of lazy writer writes may indicate that the SQL Server is running low on allocated memory .

Metric	Why it is important
Page Reads	A high number of page reads may indicate that the SQL Server is executing inefficient queries or that you should optimize the database indexes.
Page Writes	A high number of page writes may indicate that the SQL Server is executing inefficient queries or that you should optimize the database indexes.
Read ahead reads	A high number of read ahead reads may indicate that you should tune your query execution plan or that you should defragment the disk.

Available alerts

- [Average Disk Milliseconds Per Read Alert](#)
- [Average Disk Milliseconds Per Write Alert](#)
- [Disk Reads \(Per Second\) Alert](#)
- [Disk Writes \(Per Second\) Alert](#)

File Activity panel

The File Activity panel displays the top five database files with the highest relative activity since the last refresh. Available metrics allow you to view the top Reads/sec, Writes/sec, and Transfers/sec activity for all files or a selected database, disk, or file. Unexpected spikes in the amount of file activity may warn you of ineffective or excessive indexing or out-of-date statistics. For additional information, see [View file activity](#).

I/O stall chart (ms/sec)

The **I/O Stall ms/sec** chart displays the total time per second (in milliseconds) that the SQL Server processes were waiting on an I/O to complete. Chronically high I/O stall metrics may indicate a disk bottleneck.

Reads chart (per second)

The **Reads/sec** chart displays the number of physical reads per second the disk sub-system is performing to bring SQL Server database pages into the data (buffer) cache. If the reads rate is excessive for a long period of time, consider taking action to avoid performance issues.

Reads KB chart (per second)

The **Reads KB/sec** chart displays the size of the physical reads per second the disk sub-system is performing to bring SQL Server database pages into the data (buffer) cache.

Total KB chart (per second)

The **Total KB/sec** chart displays the total size of activity per second across all databases on the monitored instance.

Writes chart (per second)

The **Writes/sec** chart displays the number of physical database writes per second the disk sub-system is performing. If the writes rate is excessive for a long period of time, consider taking action to avoid performance issues.

Writes KB chart (per second)

The **Writes KB/sec** chart displays the size of the physical database writes per second the disk sub-system is performing.

Transfers chart (per second)

The **Transfers/sec** chart displays the total number of transfers per second across all databases on the monitored instance. Transfers are the sum of reads and writes. A higher transfer rate indicates a higher risk of resource issues.

When the File Activity panel displays no data

There are circumstances under which the File Activity panel does not display any data. For example, when you lose connection to the Collection Service or SQL Diagnostic Manager Repository. Another reason is the lack of collection data points. On new instances, file activity collection occurs once per hour by default. By increasing the length of time shown on the desktop client views, you may be able to see more data points. You can modify the timeframe of data displayed in SQL Diagnostic Manager by clicking **Tools > Console Options** while in the File Activity view and changing the **Keep data for the last** and **Show data for the last** values.

You also can change the database collection interval and grooming options to help you see more data points. Use the **Collect and alert on database metrics** field in the [General tab of the Monitored SQL Server Properties window](#) to manage your database collection. For grooming options, use the **Groom Sessions, Queries, Deadlocks, Waits, and History Browser data older than** field in the [Grooming Options](#) window.

- ✔ Decreasing the collection interval and grooming your data less often can result in performance issues and require additional storage space in your Repository.

Activity for a new monitored SQL Server instance

The File Activity panel displays results based on activity since the last refresh. You must have two collections to create a point on the graph, three collections to create a line. By default, the database collection interval is set at one hour, which means that the File Activity panel does not show a point until an hour after adding the instance unless you perform a manual refresh. You can refresh and collect data by clicking the **Refresh** button in the SQL Diagnostic Manager Management Console menu bar.

Activity when using the History Browser

The History Browser allows you to view the state of your SQL Server instance at the time a snapshot occurs, which is every six minutes by default. When using the History Browser with the File Activity panel, note that data may appear in one snapshot but not in others because no file activity occurred during that interval. Remember that file activity data does not appear for the first two hours of monitoring a SQL Server instance because the metrics are based on the activity since the last database refresh.

Lock Waits panel

The Lock Waits panel tracks the total number waits caused when a task is waiting to acquire a lock on your monitored SQL Server. Unexpected spikes in the number of lock waits may warn you about the beginning of a serious performance issue. Chronically high lock wait numbers may indicate the need for server maintenance, query tuning, or index updates to better handle the ongoing workload. For additional information, see [Analyze locked sessions](#).

Lock Waits chart

The Lock Waits chart displays the number of waits and the request type occurring.

Metric	Why it is important
Database	Indicates the time waiting for a database lock.
Extent	Indicates the time waiting for an extent lock.
File <i>SQL Server 2005 and above only</i>	Indicates the time waiting for a file lock.
HoBT <i>SQL Server 2005 and above only</i>	Indicates the time waiting for a Heap or BTree lock.
Key <i>SQL Server 2000 only</i>	Indicates the time waiting for a key lock.
Key/RID <i>SQL Server 2005 and above only</i>	Indicates the time waiting for a key or RID lock.
Metadata <i>SQL Server 2005 and above only</i>	Indicates the time waiting for a metadata lock.
Object <i>SQL Server 2005 and above only</i>	Indicates the time waiting for an object lock.
Page	Indicates the time waiting for a page lock.
RID <i>SQL Server 2000 only</i>	Indicates the time waiting for a relative identifier (RID) lock.
Table <i>SQL Server 2000 only</i>	Indicates the time waiting for a table lock.

Memory panel

The Memory panel tracks the usage and availability of memory on your monitored SQL Server computer. Unexpected spikes in memory usage can warn you about the beginning of a serious performance issue. Chronically high memory metrics may indicate the need for server maintenance, query tuning, or other diagnostics to better handle the ongoing workload. Use the **Memory** view on the **Resources** tab to track key memory statistics over a period of time. For additional information, see [Get memory performance details](#).

SQL Server Usage chart

The **SQL Server Usage** chart compares the amount of memory allocated and used by SQL Server to the total memory consumed by a computer.

Metric	Why it is important
SQL allocated	Indicates how much available memory is allocated to SQL Server from the operating system (SQLOS). Compare this metric to how much memory is actually being used by SQL Server.
SQL used	Indicates how much memory is currently used by SQL Server from the total allocated (SQL allocated) to this application. A higher SQL memory percentage indicates that you may have poorly performing queries or a fragmented database.
Total used	Indicates how much physical memory is used by the operating system (SQL Server and other applications). A higher total memory percentage indicates that the overall workload of the SQL Server computer is too intensive for the available resources.

SQL Server Paging chart

The **SQL Server Paging** chart plots the number of swapped pages per second, over time, as calculated by the [memory paging metric](#).

VM Memory Usage chart

The **VM Memory Usage** chart compares the amount of memory allocated and used by SQL Server to the total memory consumed by this virtual machine. For additional information about how SQL Diagnostic Manager works with virtual machines and collects metrics, see [How SQL Diagnostic Manager works with a virtual environment](#).

Metric	Why it is important
Active	Indicates the amount of memory actively used by this virtual machine.
Ballooned	Indicates the amount of memory temporarily reclaimed by the host system for use in other areas. A higher ballooned value indicates that the host server is under memory pressure. Review your load balancing for the host server.

Metric	Why it is important
Consumed	Indicates the amount of memory consumed by this virtual machine. This value does not include shared memory or overhead memory for this VM.
Granted	Indicates the amount of physical memory granted by the host to this virtual machine.
Swapped	Indicates the amount of memory swapped to disk by the hypervisor.

Host Memory Usage chart

The **Host Memory Usage** chart compares the amount of memory allocated and used by SQL Server to the total memory consumed on this host server. For additional information about how SQL Diagnostic Manager works with virtual machines and collects metrics, see [How SQL Diagnostic Manager works with a virtual environment](#).

Metric	Why it is important
Active	Indicates the amount of memory actively used by all of your VMs on this host server.
Ballooned	Indicates the amount of memory temporarily reclaimed by all of your VMs on the host system for use in other areas. A higher ballooned value indicates that the host server is under memory pressure. Review your load balancing for the host server.
Consumed	Indicates the amount of memory consumed by all of your VMs on this host server. This value does not include shared memory or overhead memory for these VMs.
Granted	Indicates the amount of physical memory granted by the host to all of your VMs.
Swapped	Indicates the amount of memory swapped to disk by the hypervisor.

Available alerts

- [Host Memory Usage \(Percent\) Alert](#)
- [O/S Memory Usage \(Percent\) Alert](#)
- [O/S Paging Alert](#)
- [SQL Memory Usage \(Percent\) Alert](#)
- [VM Memory Usage \(Percent\)](#)

Network panel

The Network panel tracks the performance of the network connection being used by your monitored SQL Server. Unexpected spikes in packet rates and response times may warn you about the beginning of a serious performance issue. Chronically high network metrics may indicate excessive network traffic or a high server workload, which may require you to move some monitored instances to other computers to free up network resources and balance the workload.

SQL Server Throughput chart

The **SQL Server Throughput** chart plots how many packets this SQL Server sends and receives over time.

Metric	Why it is important
Packets received/sec	A consistently high rate may indicate that the packet size is too small.
Packets sent/sec	A consistently high rate may indicate that the packet size is too small.

VM Network Usage Throughput chart

The **VM Network Usage Throughput** chart plots how much data this virtual machine transmits and receives in KB/second over time. For additional information about how SQL Diagnostic Manager works with virtual machines, see [How SQL Diagnostic Manager works with a virtual environment](#).

Metric	Why it is important
Transmitted KB/sec	A consistently high rate may indicate that data throughput is saturated. Investigate load balancing your network traffic.
Received KB/sec	A consistently high rate may indicate that data throughput is saturated. Investigate load balancing your network traffic.

Host Network Usage Throughput chart

The **Host Network Usage Throughput** chart plots how much data this host server transmits and receives in KB/second over time. For additional information about how SQL Diagnostic Manager works with virtual machines, see [How SQL Diagnostic Manager works with a virtual environment](#).

Metric	Why it is important
Transmitted KB/sec	A consistently high rate may indicate that data throughput is saturated. Investigate load balancing your network traffic.
Received KB/sec	A consistently high rate may indicate that data throughput is saturated. Investigate load balancing your network traffic.

Response Time gauge

The **Response Time** gauge displays the time (in milliseconds) SQL diagnostic manager currently needs to send a simple SQL command to the SQL Server instance, have it processed, and receive the returned result set. This value corresponds to the [SQL Server Response Time](#) metric.

Available alerts

- [SQL Server Response Time Alert](#)

Server Waits panel

The Server Waits panel tracks the total number and time spent on waits affecting your monitored SQL Server. Unexpected spikes in the number of server waits may warn you about the beginning of a serious performance issue. Chronically high server wait times may indicate the need for server maintenance, query tuning, or index updates to better handle the ongoing workload. For additional information, see [View server waits](#).

- ✔ Server Waits chart data is not available for monitored servers using SQL Server 2000. For SQL Server 2000 users, SQL Diagnostic Manager monitors [lock waits](#).

Server Waits chart

The Server Waits Chart displays the overall wait time for key areas on your monitored SQL Server instance.

Metric	Why it is important
I/O	Indicates the amount of input and output occurring on your SQL Server instance. A consistently high rate may indicate a higher risk of performance issues as the SQL Server processes wait for I/O to become available.
Locks	Indicates the wait time caused by locks on databases hosted by this SQL Server instance.
Log	Indicates the wait time related to database transaction logs.
Memory	Indicates all wait time related to memory operations.
Signal	Indicates that a SQL Server process is waiting for CPU time after the associated resources became available for processing.
Other	Indicates all lock waits not included in other categories and not excluded from monitoring.

Sessions panel

The Sessions panel tracks the performance of the active and blocked sessions running on your monitored SQL Server. Unexpected spikes in the number of concurrent blocked sessions may warn you about the beginning of a serious performance issue. Chronically high blocked and deadlocked sessions are a significant performance concern. For additional information, see [Get sessions performance summary](#).

Activity & Blocking chart

The Activity and Blocking chart shows the total number of active sessions on your monitored SQL Server instance and any blocking or deadlocks that may occur between sessions.

Metric	Why it is important
Active	Indicates the number of active sessions on the SQL Server instance.
Blocked Sessions	Indicates the number of sessions on the SQL Server instance being blocked by other sessions holding requested locks.
Lead Blockers	Indicates sessions that block at least one session, which can then block other sessions.
Total Deadlocks	Indicates the total number of deadlocks on the SQL Server instance. A deadlock has no lead blocker and is circular in nature.

Client Computers gauge

The Client Computers gauge displays the total number of unique client computers connected to the SQL Server instance.

Available alerts

- [Blocked Sessions Alert](#)
- [Client Computers Alert](#)
- [Total Deadlocks Alert](#)

Tempdb panel

The Tempdb panel tracks the status of the tempdb database on your monitored SQL Server instance. Unexpected spikes in the use of tempdb usage and space may warn you to allocate additional space to prevent tempdb issues from causing a performance issue on your server. For additional information, see [how to get the tempdb status summary](#).

✔ Problems with tempdb can cause an entire SQL Server instance to become inoperable.

Tempdb Space Used chart

The Tempdb Space Used chart provides different views of how your database is used over time based on the object type. Tempdb can fill up quickly when you are low on disk space and or have a low maximum size for database growth. In environments where read committed snapshot isolation is in use, long running transactions may also cause rapid growth in tempdb.

Metric	Why it is important
Internal	Indicates the amount of tempdb space used by internal objects.
Mixed	Indicates the amount of tempdb space used by mixed extents.
User	Indicates the amount of tempdb space used by user objects.
Version Store	Indicates the amount of tempdb space used by the version store.

Tempdb Contention chart

The Tempdb Contention chart provides the latch wait time for the allocation pages of your tempdb. These latch waits are associated with performance degradation for the related queries. Latch contention is usually an indication that you should create additional tempdb data files or if you already have multiple files, make adjustments so they are equally sized. For more information about tempdb contention, see [Monitor your tempdb database](#).

Metric	Why it is important
GAM Page	Indicates the latch wait time for the global allocation map pages of tempdb.
PFS Page	Indicates the latch wait time for the page free space pages of tempdb.
SGAM Page	Indicates the latch wait time for the shared global allocation map pages of tempdb.

Version Store Cleanup Rate

The **Version Store Cleanup Rate** gauge displays the current version store cleanup rate and the version generation rate of the data rows necessary to support snapshot isolation. Maintaining a cleanup rate above or equal to the generation rate helps you avoid filling up tempdb. Note that long-running snapshot isolation transactions prevent cleanup of older entries in the tempdb version store, causing growth which can affect performance and cause tempdb to run out of space. When interpreting this gauge, note that version store cleanup occurs only once per minute while generation may be ongoing. For more information about tempdb version store, see [Monitor your tempdb database](#).

Available alerts

- [Tempdb Contention Alert](#)
- [Version Store Generation Ratio Alert](#)
- [Version Store Size Alert](#)

Virtualization panel

The Virtualization panel tracks the performance of your virtual machine or host, depending on the selection in the panel. Unexpected spikes in virtual memory or disk usage may warn you about the beginning of a serious performance issue indicating the need for server maintenance to better handle the ongoing workload. For additional information about how SQL Diagnostic Manager works with virtual machines and collects metrics, see [How SQL Diagnostic Manager works with a virtual environment](#).

- ✔ Note that you can customize your Dashboard view to include two Virtualization panels and add one for your virtual machine and the other for your host.

Virtual Memory Usage chart

The Virtual Memory Usage chart displays the amount of memory allocated and used by SQL Server to the total memory consumed on this VM or host server. The **Resources > Memory** view allows you to track your usage over a period of time, along with other key memory statistics. For additional information about how SQL Diagnostic Manager works with virtual machines and collects metrics, see [How SQL Diagnostic Manager works with a virtual environment](#).

Metric	Why it is important
Active	Indicates the amount of active memory used by this virtual SQL Server instance.
Ballooned	Indicates the amount of ballooned memory on this virtual SQL Server instance. A spike in this value alerts you of the low memory status of the host.
Consumed	Indicates the amount of memory consumed by this virtual SQL Server instance.
Granted	Indicates the amount of memory granted to this virtual SQL Server instance.
Swapped	Indicates the amount of swapped memory on this virtual SQL Server instance. A spike in this value alerts you of the low memory status of the host.

CPU Ready Wait Time gauge

The CPU Ready Wait Time gauge displays the current gauge value of the vCPU Ready Time metric available on the Server Details view. Use this metric to determine the length of time that virtual CPU in your virtual machine is waiting on a physical CPU on your host to become available. A spike in this value may indicate over committed physical CPUs on the host server.

Virtual Disk Usage chart

The Virtual Disk Usage chart displays the amount of data read from and written to disk for the VM or host server during the sample interval. The **Resources > Disk** view allows you to track your usage over a period of time, along with other key disk statistics.

Metric	Why it is important
Disk Read	A high number of disk reads may indicate that the VM or host server is executing poorly-performing queries.
Disk Write	A high number of disk writes may indicate that the VM or host server is running low on allocated memory.

Available alerts

- [Host Memory Swap Detected Alert](#)
- [VM CPU Ready Wait Time \(ms\) Alert](#)
- [VM Memory Swap Detected Alert](#)
- [VM Reclaimed/Ballooned Memory \(KB\) Alert](#)

Get the SQL Server performance details

The Details view displays a list of properties for the selected SQL Server instance, a list of the collected metrics and their characteristics, and provides a chart for you to graphically display metrics selected from the displayed list. For each metric, SQL Diagnostic Manager displays the current value, value range for reference, alert status, metric category, thresholds for each alert, and whether the metric is a custom counter. To add a metric to the Details chart, check the check box in the chart column for the appropriate metric.

- ✔ All fields highlighted in the **State** column are associated with alerts. Right-click any of these metrics and select **Configure Alerts** to edit the associated thresholds.

Access the Details view

You can open the Details view of the SQL Diagnostic Manager Overview tab by selecting the appropriate SQL Server instance, and then clicking **Overview > Details**.

SQL Server instance properties area

The following properties are displayed for the selected monitored SQL Server instance.

Version

Indicates the current SQL Server version including service pack and build information on this SQL Server instance.

Edition

Indicates the current SQL Server edition, including Enterprise, Business Intelligence, or Standard, installed on this SQL Server instance.

Running

Indicates the timestamp associated with the most recent restart of this SQL Server instance.

Clustered

Indicates whether this SQL Server instance is part of a cluster. If it is part of a cluster, SQL Diagnostic Manager displays the name of the active node.

Processors

Indicates the total number of processors associated with this SQL Server instance and how many of these processors are currently in use.

Host

Indicates the name of the server hosting this SQL Server instance.

Host OS

Indicates the current operating system on the server hosting this SQL Server instance.

Host Memory

Indicates the total amount of memory on the server hosting this SQL Server instance.

Databases

Indicates the current number of databases on this SQL Server instance.

Data Size

Indicates the current size of the data on this SQL Server instance.

Log Size

Indicates the current size of the log files on this SQL Server instance.

VM Name

Indicates the name of the virtual machine on which this monitored SQL Server instance is running.

Virtualization Host Server

Indicates the name or IP address of the virtualization host server on which this virtual machine is running.

Host Memory

Indicates the total amount of memory on the host server on which this virtual machine is running.

Host Physical Processors

Indicates the total number of physical processors associated with the host server on which this virtual machine is running.

Host Logical Processors

Indicates the total number of logical processors associated with the host server on which this virtual machine is running. The number of logical processors is different from the number of physical processors only if the CPU on the host server supports Hyper-Threading.

Understanding your chart data and refresh interval

SQL Diagnostic Manager charts provide data whether collected real time or as part of a scheduled collection event. For additional information about how SQL Diagnostic Manager displays your chart data, see [Charts](#).

Monitor SQL Server configuration settings

The Configuration view allows you to view the SQL Server configuration settings for the selected SQL Server instance. When diagnosing problems with SQL Server performance it is helpful to see how it is configured. In addition, SQL Diagnostic Manager offers the ability to edit some configuration settings, depending on the option selected.

Access the Configuration view

You can open the Configuration view of the SQL Diagnostic Manager Overview tab by selecting the appropriate SQL Server instance, and clicking **Overview > Configuration**.

To edit a configuration setting:

1. Select an option from the list.
2. Click the **Edit Value** hyperlink in the Details pane.
3. Enter the new value.
4. Click **OK**.

Edit your configuration value

The Edit Configuration Value window allows you to enter a new value for the selected SQL Server instance configuration option. Any change to the selected value does not take effect until you restart the selected SQL Server instance.

Access the Edit Configuration Value window

You can access the Edit Configuration Value window by selecting an option, and then click the **Edit Value** option located in the Details pane.

View active alerts for this instance

The Active Alerts view allows you to view a list of all the active alerts as of the last data collection. Click any alert in the list to view details at the bottom of the window.

Access the Active Alerts view

You can open the Active Alerts view of the SQL Diagnostic Manager Overview tab by selecting the appropriate SQL Server instance, and clicking **Overview > Active Alerts**.

Available actions

The Active Alerts view includes the [12-Hour Forecast](#) option to display the likelihood an alert is generated at a given time during the day, broken down into two-hour segments. This information helps you know on which areas you should concentrate your focus to avoid problems within your SQL Server environment.

In addition to providing the ability to see all your active alerts in a single window, the Active Alerts view allows you to perform the following functions via the right-click context menu for each alert:

- Open the Real Time view for more information on the metric that caused the alert.
- Open the Historical view to review the history of the metric that caused the alert. The **Show Historical View** option is not available for all alerts.
- Open the Show Details view for more information on the metric that caused the alert. The **Show Details View** option is not available for all alerts.
- Open the SQL Diagnostic Manager online help to the topic for the selected alert.
- Clear the selected alert or clear all alerts of the selected type for this SQL Server instance. The **Clear Alert** and **Clear All Alerts of this Type for this Instance** options are not available for all alerts.
- Open the [alert configuration for the metric](#).
- [Snooze the alert](#) for a set period of time.
- Collapse or expand all of the displayed alert groups. This option is available only when you use the **Group By Box** functionality.
- You can also print, export results to Microsoft Excel, and copy the alert information to your Clipboard for additional use.




Get a 12-hour forecast of alert activity

Using current and historical data, SQL Diagnostic Manager uses statistical analysis to estimate the likelihood that an event generates a particular alert. This forecasting lets you proactively focus on certain areas in your SQL Server environment and prevent problems from occurring.

SQL Diagnostic Manager begins building forecasting models once you receive three critical or warning alerts for a given metric.

The 12-hour Forecast is separated into two-hour segments. Each two-hour segment displays the chance of an alert occurrence. Click the associated percentage chance to see the [alert forecast details](#). This window displays the alerts that may occur over the two-hour time segment.

12 Hour Forecast icons

Icon	Description
	OK - There is a low chance (less than 20%) that a warning or critical alert occurs in the two-hour time segment leading to the forecast time.
	Warning - There is a moderate likelihood (less than 50%) that a critical alert occurs, or a high likelihood (50% or higher) that a warning alert occurs, in the two-hour time segment leading to the forecast time.
	Critical - There is a very high likelihood (50% or higher) that a critical alert occurs in the two-hour time segment leading to the forecast time.

Alert forecast details

The Alert Forecast Details window displays the Server, Alert Metric, the likelihood the alert occurs, and the type of alert expected in the two-hour window leading up to this forecast time. To sort the list, click any column heading.

Snooze alerts

You have the ability in SQL Diagnostic Manager to snooze or pause all alert generation for a specified period of time. For example, you can snooze a specific alert through the Active Alerts tab. Right-click the alert that you no longer want to view, and then select **Snooze Alert**.

If you want to pause multiple alerts at the same time, you can mass snooze alert generation. To mass snooze, right-click a specific instance, and select **Snooze Alerts**. This option is also accessible when right-clicking a specific tag, using certain server views, and when using the Tools menu.

This feature is important if you are working on known issues and you want to avoid unnecessary alert traffic.



The **Resume Alerts** option allows you to resume any alerts set to snooze.

View past performance

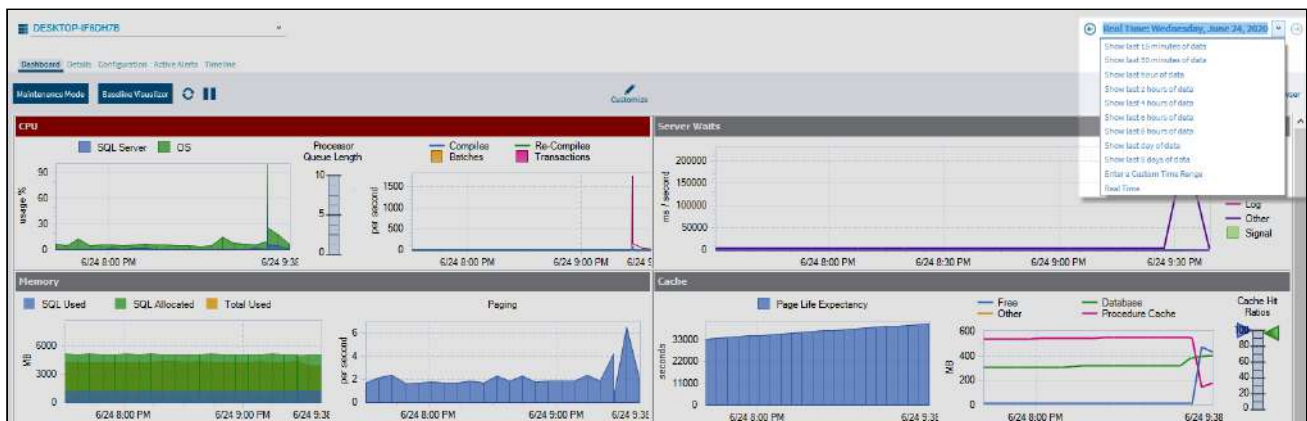
The History Browser is intended to provide information pertaining to the state of your SQL Server instance at the time a standard snapshot is taken. You can use this information to diagnose and resolve issues to keep the issue from happening again. The History Browser allows you to select a historical snapshot collected by the standard refresh and view the data that was collected using almost all the SQL Diagnostic Manager real-time views.

i By default, Session details and inactive alerts are stored for 31 days while standard metric information is stored for 365 days. You can edit these options on the [Grooming Options](#) window.

Historical Snapshot drop-down

SQL Diagnostic Manager has a historical snapshot drop-down in the top-right corner, the drop-down allows you to retrieve faster historical and real time data. The following options are available for the data retrieval:

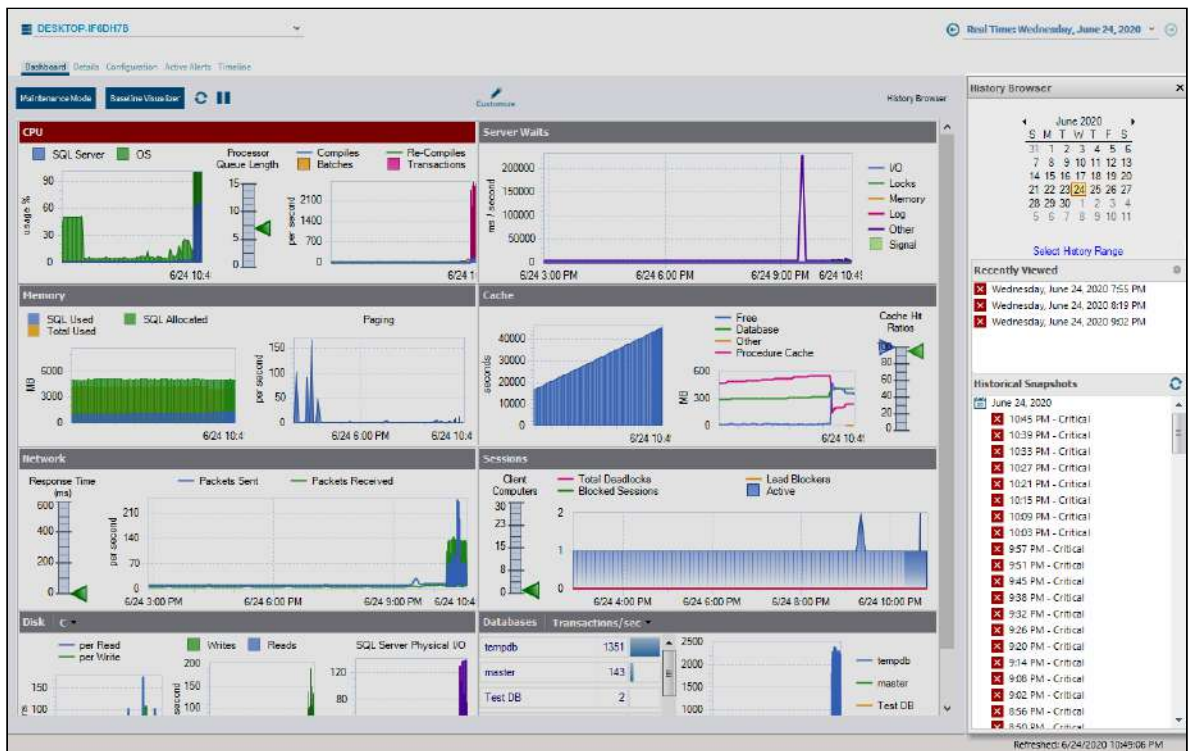
- Real Time
- Show last 15minutes of data
- Show last 30 minutes of data
- Show last hour of data
- Show last 2 hours of data
- Show last 4 hours of data
- Show last 6 hours of data
- Show last day of data
- Show last 5 days of data
- Enter a Custom Time Range



Select a specific snapshot in time in the History Browser

To select a specific snapshot:

1. Open the History Browser.
2. Select the date from the calendar. You can also select the time range to filter the number of snapshots displayed.
3. Select the corresponding snapshot from the Historical Snapshots list.



✔ The icons next to each of the snapshots represent the highest level of alert thresholds violated. These icons can help you identify a specific snapshot. In addition, you can hover over snapshots in the list to see the most critical alerts associated with the snapshot.

⚠ The alert status listed by snapshots in the History Browser may not reflect the latest status due to delays in writing history to the repository. This status corrects eventually in the History Browser.

Alerts with associated views that support historical snapshots

On the Alerts view, some of the alerts allows you to retrieve the associated view at the time the alert was activated. To access this view, right-click the alerts, and the click **Show Historical View** in the **Details** pane.

The following table displays a list of alerts that provide an associated view in SQL Diagnostic Manager.

Alert	Associated View
Average Disk Milliseconds Per Read	Resources
Average Disk Milliseconds Per Transfer	Resources
Average Disk Milliseconds Per Write	Resources
Blocked Sessions (Count)	Sessions

Alert	Associated View
Blocking Session Wait Time (Seconds)	Sessions
Client Computers (Count)	Sessions
Databases Read/Write Errors Occurred	Databases
Disk Reads Per Second	Resources
Disk Transfers Per Second	Resources
Disk Writes Per Second	Resources
Host CPU Usage (Percent)	Resources
Host Memory Usage (Percent)	Resources
Oldest Open Transactions (Minutes)	Sessions
OS Average Disk Queue Length (Count)	Resources
OS Average Disk Queue Length Per Disk (Count)	Resources
OS Disk Time (Percent)	Resources
OS Disk Time Per Disk (Percent)	Resources
OS Memory Usage (Percent)	Resources
OS Paging (Per Second)	Resources
OS Privileged Time (Percent)	Resources
OS Processor Queue Length (Count)	Resources
OS Processor Time (Percent)	Resources
Page Life Expectancy	Server

Alert	Associated View
Session CPU Time (Seconds)	Sessions
SQL Server CPU Usage (Percent)	Resources
SQL Server Memory Usage (Percent)	Resources
SQL Server Response Time (Milliseconds)	Sessions
Tempdb Contention (ms)	Databases
User Connections (Percent)	Sessions
VM CPU Usage (Percent)	Resources
VM Memory Usage (Percent)	Resources

Views that support historical snapshots

Users may notify you of SQL Server problems that have happened in the past. Using the History Browser, you can go back to a point in time and view the state of your system and also compare system states at different points in time.

The following views support historical snapshots:

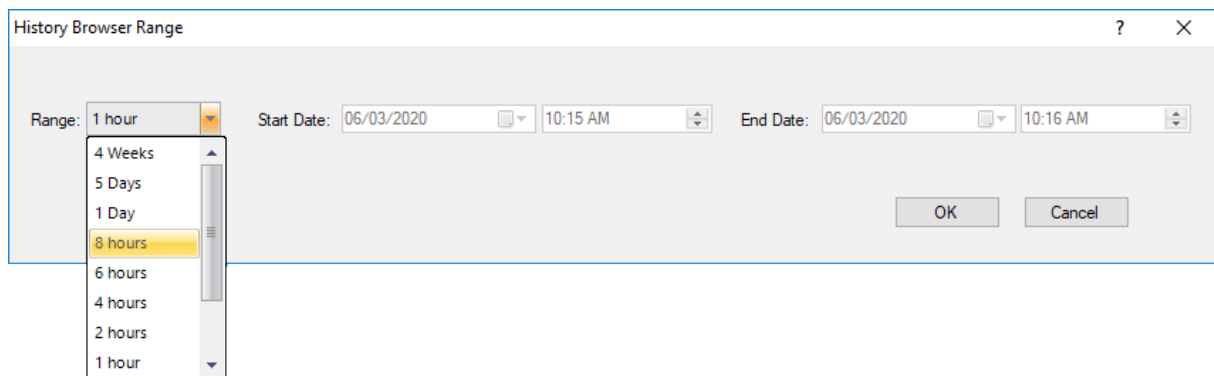
- Overview > Dashboard
- Overview > Details
- Overview > Active Alerts
- Sessions > Summary
- Sessions > Details
- Sessions > Locks
- Sessions > Blocking
- Queries > Signature Mode
- Queries > Statement Mode
- Queries > Query History
- Queries > Query Waits
- Resources > Summary
- Resources > CPU
- Resources > Memory
- Resources > Disk
- Resources > File Activity
- Resources > Server Waits
- Databases > Tempdb Summary
- Databases > Availability Group
- Analyze > History

History Browser Range

Find the History Browser Range access in the Web Console, in the History Browser of the Desktop Client console, and when you select 'Enter a Custom Time Range' from the Historical Snapshot drop-down.


Follow these steps to configure the History Browser Range:


1. In the desktop console, click **History Browser**.
2. Under the calendar select the option **Select History Range**.
3. The History Browser Range window opens.
4. Select the range from the drop down list scale, these options are the following: 4 Weeks, 5 Days, 1 Day, 8 hours, 6 hours, 4 hours, 2 hours, 1 hour, 30 minutes, 15 minutes, and Custom.
5. When selecting **Custom**, the user needs to select a Start Date/Time and End Date/Time.
6. Click **OK**.



Pause alerts and decrease collection

Maintenance Mode allows you to temporarily stop alert generation and the collection of performance metrics for the selected SQL Server instances. Using Maintenance Mode on one instance does not prevent the collection of performance metrics and notifications for those instances that are not in Maintenance Mode.

 If you open a SQL Diagnostic Manager dashboard for an instance that is in Maintenance Mode, real-time data collection will be attempted.

 It is possible to enable or disable maintenance mode on several SQL Server instances, see [About Mass Maintenance Mode](#).

Uses for putting a SQL Server instance into Maintenance Mode

Enable Maintenance Mode before you begin any regularly scheduled maintenance or testing on a monitored SQL Server instance. You can also enable Maintenance Mode when you need to bring a SQL Server instance offline for unplanned diagnostics.

Be aware that the total number of monitored SQL Server instances allowed by your license includes monitored instances that are in Maintenance Mode.

Bring the SQL Server back online in SQL Diagnostic Manager

Disabling Maintenance Mode allows SQL Diagnostic Manager to resume performance statistics collection and alert generation for the selected SQL Server instance. Before you enable Maintenance Mode, make sure you complete all maintenance and testing on the target SQL Server instance.

If you have a SQL Server instance in Maintenance Mode and are ready to bring it back online in SQL Diagnostic Manager, click **Maintenance Mode** on the Server Overview tab.

Configure SQL Diagnostic Manager monitoring properties

The Monitored SQL Server Properties window allows you to edit the SQL Diagnostic Manager monitoring settings for the registered SQL Server instance. To access the Monitored SQL Server Properties window, right-click a SQL Server instance from the **Server** tree and select **Properties**.

The Monitored SQL Server Properties window includes the following tabs:

General

The [General](#) tab displays common properties such as the data collection interval, credentials, and encryption options.

Baseline Configuration

The [Baseline Configuration](#) option allows you to set the date range and time used to calculate a performance baseline for the monitored SQL Server instance.

Query Monitor

Allows you to enable or disable query monitoring, and set the options for [Query Monitor](#).

Activity Monitor

Allows you to enable or disable monitoring of non-query activities, and set the options for [Activity Monitor](#).

Replication

The [Replication](#) tab lets you disable replication statistics collection to conserve system resources.

Table Statistics

The [Table Statistics](#) tab allows you to select the time and days SQL Diagnostic Manager collects table statistics on your SQL Server instance. In addition, you can select the minimum size table needed for SQL Diagnostic Manager to rebuild the index.

Custom Counters

Allows you to [link custom counters to the monitored SQL Server instance](#).

Maintenance Mode

The [Maintenance Mode](#) tab allows you to enter the maintenance schedule for your server. This allows you to turn off alerts and notifications for the time periods when you know the server is unavailable.

OS Metrics

The [OS Metrics](#) tab allows you to select how you want your operating system data collected, whether through OLE automation or direct WMI. You can also choose to not collect OS data.

Disk Drives

The [Disk Drives](#) tab allows SQL Diagnostic Manager to alert on disk drives and database space calculations for the monitored SQL Server instance.

Cluster Settings

The [Cluster Settings](#) lets SQL Diagnostic Manager trigger an alert whenever a cluster is running on a non-preferred node within a clustered environment.

Wait Monitoring

The [Wait Monitoring](#) allows you to collect and view wait statistics in the hopes of diagnosing any bottleneck on your system.

Virtualization

The [Virtualization](#) displays the name of the host server on which the selected monitored SQL Server instance is running.

Analysis Configuration

Allows you to schedule a prescriptive analysis of your monitored SQL Server instance. For more information, visit [Configure prescriptive analysis options](#).

Set general server options

The General tab of SQL Diagnostic Manager allows you to edit the most common property settings such as:

- [Select the type of server](#)
- [Associate tags with an instance](#)
- [Select a diagnostic data collection interval](#)
- [Assign a friendly name to your SQL Server instance](#)
- [Collect extended session data](#)
- [Assign data collection credentials](#)
- [Select encryption options](#)

Access the General tab

You can access the General tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**.

Monitored SQL Server Properties - DESKTOP-9EFPMF4

General

- Baseline Configuration
- Query Monitor
- Activity Monitor
- Replication
- Table Statistics
- Custom Counters
- Maintenance Mode
- OS Metrics
- Disk Drives
- Cluster Settings
- Wait Monitoring
- Virtualization
- Analysis Configuration

Change the most common properties for this monitored SQL Server.

What type of server is this?

Which tags should be associated with this SQL Server?

How often should diagnostic data be collected for this SQL Server?

Collect diagnostic data and raise alerts (max 30 minutes):

Alert if the server is inaccessible (max 10 minutes):

Collect and alert on database metrics (max 24 hours):

Want to give a friendly name for this SQL Server?

Would you like to collect extended session data for this SQL Server?

Collect extended session data, including session details, locks and blocks

Limit executions of DBCC Inputbuffer to:

What credentials should be used when collecting data from this SQL Server?

Windows Authentication

SQL Server Authentication

Login name:

Password:

Advanced Encryption Options

Encrypt Connection (SSL)

Trust Server Certificate (bypass certificate validation)

Select the type of the monitored server

The drop-down menu allows you to identify the type of server you are monitoring, select from the following options: Windows, Amazon RDS SQL Server (RDS SQL), Microsoft Azure SQL Database (AzureDB), or Linux.

Associate tags with an instance

The tag field allows you to [select a tag](#) to add your server to or [add a new tag](#). Tags help you organize server instances into meaningful groups.

Select a diagnostic data collection interval

The data collection interval is the amount of time SQL Diagnostic Manager waits between each collection of data on the selected SQL Server instance. You can set a different interval for each of the following functions:

Collect diagnostic data and raise alerts (max 30 minutes)

The interval between times that the SQLDM Collection Service collects diagnostic data and raises the associated alerts. Lower values cause SQL Diagnostic Manager to raise alerts more quickly, but also cause more frequent refreshes, which increases your monitoring overhead.

Alert if the server is inaccessible (max 10 minutes)

The server availability verification interval is the amount of time SQL Diagnostic Manager waits between verifying availability on the selected SQL Server instance. If the connection test collector does not complete within the time specified in the **Alert if the server is inaccessible (max 10 minutes)** field, the SQL Server instance is considered unresponsive. Setting this field to a very low value can result in false positive alerts.

Collect and alert on database metrics (max 24 hours)

The interval between times that the SQL Diagnostic Manager Collection Service collects database space-related data and raises the associated alerts. Lower values cause SQL Diagnostic Manager to raise alerts more quickly, but also cause more frequent refreshes, which increases your monitoring overhead. In environments with a large number of databases whose sizes do not change rapidly, setting the database data collection to a long interval can greatly reduce the monitoring footprint.

By default, SQL Diagnostic Manager collects diagnostic data every six minutes and database data every 60 minutes. Consider the following factors before selecting your data collection interval:

Data collection purpose

If your goal with this data collection is to identify broad trends and alert on critical downtime, then a longer refresh interval is appropriate. If you are closely monitoring your SQL Server instance for minute-by-minute changes, use a shorter duration.

Data collection frequency

When diagnosing specific problems or when working with a problematic SQL Server instance, a short interval lets you capture enough data to diagnose the issue. In most situations, the default interval provides sufficient data for your diagnostic needs. Note that SQL Server instances used only occasionally may require less monitoring attention.

Notification of existing or potential problems

Remember that a lower collection interval results in SQL Diagnostic Manager raising alerts more quickly, but also increases your monitoring overhead. This is true when any setting causes more frequent refreshes.

System resource impact

Although SQL Diagnostic Manager limits the amount of system resource impact when collecting data, short collection times on SQL Server instances with large amounts of data could potentially cause

system performance degradation. Using the default interval should meet your data collection needs while limiting any system resource impacts.

Space needed for the SQL Diagnostic Manager Repository

A lower collection interval results in more frequent refreshes which are all stored in the SQL Diagnostic Manager Repository. Make sure you have enough available space to accommodate these lower settings.

Data spikes


SQL Diagnostic Manager averages most metrics over the timeframe between collections. More-frequent data collection causes increased movement in averages because of short-duration events. For example, a 15-second CPU spike has a greater effect on a one-minute refresh than on a six-minute refresh. You can use alert smoothing to reduce the impact of data spikes on a per-alert basis.

Assign a friendly name to your SQL Server instance

SQL Diagnostic Manager allows you to assign a friendly name to your monitored SQL Server instance that is meaningful to you and easy to identify. SQL Diagnostic Manager displays the friendly name in the server tree of the desktop client and in the web console.

Collect extended session data

The extended session collection data includes important session information, such as details, locks, and blocks. If this information is important to you, make sure the **Collect extended session data, including session details, locks, and blocks** check box is selected.

 The extended session collection data gathers all the executing sessions, the idle sessions and all the outstanding locks; regardless of their blocking status. This information includes tempdb usage data which is used to populate the session list in the tempdb view.

Limit the number of DBCC Inputbuffer executions

You can limit the number of executions performed by the DBCC Inputbuffer, which retrieves the actual input command for the Session Details view, among others. Note that on busy servers, decreasing the **Limit executions of DBCC Inputbuffer to** value can reduce monitoring impact.

Data collection credentials

SQL Diagnostic Manager uses the specified credentials to collect data from the monitored SQL Server instance. You can choose to use either Windows authentication or SQL Server authentication.

Windows authentication

Windows Authentication uses the security of the operating system to create a trusted connection only if the account matches a security account defined in SQL Server. This security account must have sufficient permissions on the monitored instance to collect data and OS metrics.

SQL Server authentication

Select this option to use the credentials of a specific SQL Server account.

Select encryption options

SQL Diagnostic Manager allows you to designate encryption methods used to encrypt data between the Collection Service and the monitored SQL Server instance. You can choose to use SSL or SSL with Trust Service Certificate.

Encrypt Connection (SSL)

This option sets a flag in the connection properties that is used when the collection service connects to a monitored server that specifies that SSL (Secure Sockets Layer) is used to encrypt the data between the collection service and the monitored SQL Server instance. For this option to work correctly, configure the monitored SQL Server instance to support encryption.

Trust Server Certificate (Bypass Certificate Validation)

This option is available only when the Encrypt Connection (SSL) is selected. This option allows you to skip the certificate validation when a SQL Server instance establishes a connection. If SSL on the monitored SQL Server instance is not configured to use a certificate that the collection service trusts, the connection is rejected unless the Trust Server Certificate option is selected.

Configure server baseline options

The Baseline Configuration tab of the Monitored SQL Server Properties window allows you to configure your server baseline options. SQL Diagnostic Manager calculates a baseline out of a pool of collected data based on the selected period and collection interval, and is represented by mean, standard deviation, minimum, maximum, and count statistics.

Some benefits of using this feature include:

- **Effective alert thresholds** - SQL Diagnostic Manager uses the baseline to provide alert recommendations which you can use to set effective [alert thresholds](#).
- **Effective baseline periods** - Use the [Baseline Visualizer](#) to clearly visualize when to establish baseline periods.
- **Alerting and recommendations**
- **Multiple baselines** - Define and schedule [multiple baselines](#) per server.
- **Quick propagation of baselines** - [Apply baselines to more than one instance](#).
- **Baseline display in metric graphs** - [View relevant baseline in effect](#) in various metric graphs of the Resources tab.



User accounts with the View privilege can access the Baseline Configuration window, but cannot make any changes. User accounts with the Modify privilege can make changes to the baseline configuration of their monitored SQL Server instances.

Access the Baseline Configuration tab

You can access the Baseline Configuration tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Baseline Configuration**.

Monitored SQL Server Properties - SIDROCTST02\SQL2014

General

Baseline Configuration

Query Monitor

Activity Monitor

Replication

Table Statistics

Custom Counters

Maintenance Mode

OS Metrics

Disk Drives

Cluster Settings


Wait Monitoring

Virtualization

Analysis Configuration

Configure a performance baseline for your monitored SQL Server.

What date range should be used to calculate the performance baseline for this server?

 You should select a date range that represents the typical operation of your monitored SQL Server. This baseline will be used to offer recommendations to help you better configure your alerts in the future.

Dynamic (calculates a dynamic baseline based on the last 7 days of collected data)

Custom

From: 12/25/2018 To: 1/24/2019 Manage Baseline

Which days and time period within those days should be used to calculate the baseline?

Baseline Name: Default

Sun Mon Tue Wed Thu Fri Sat

From: 08:00 AM To: 05:00 PM

Would you like to apply this baseline configuration to other SQL Servers?

Apply Baseline Configuration

OK Cancel

Select a baseline period

The baseline period you specify should represent a typical operation of the monitored SQL Server instances to which this baseline applies.

Dynamic

Calculates a dynamic baseline based on a moving window of the past seven days of collected data.

Custom

Allows you to set a specific time range that does not change.

Select days and time periods

SQL Diagnostic Manager allows you to select the date range used to calculate the performance baseline for one or more selected SQL Server instances. The day and time period fields allow you to select the days and times on which to base your alert baseline calculations. These settings apply to both the Automatic and Custom baseline selections.



When your SQL Server instances are not busy, clear the days and times to remove them from the calculation and provide a more realistic and accurate baseline.

Add multiple baselines

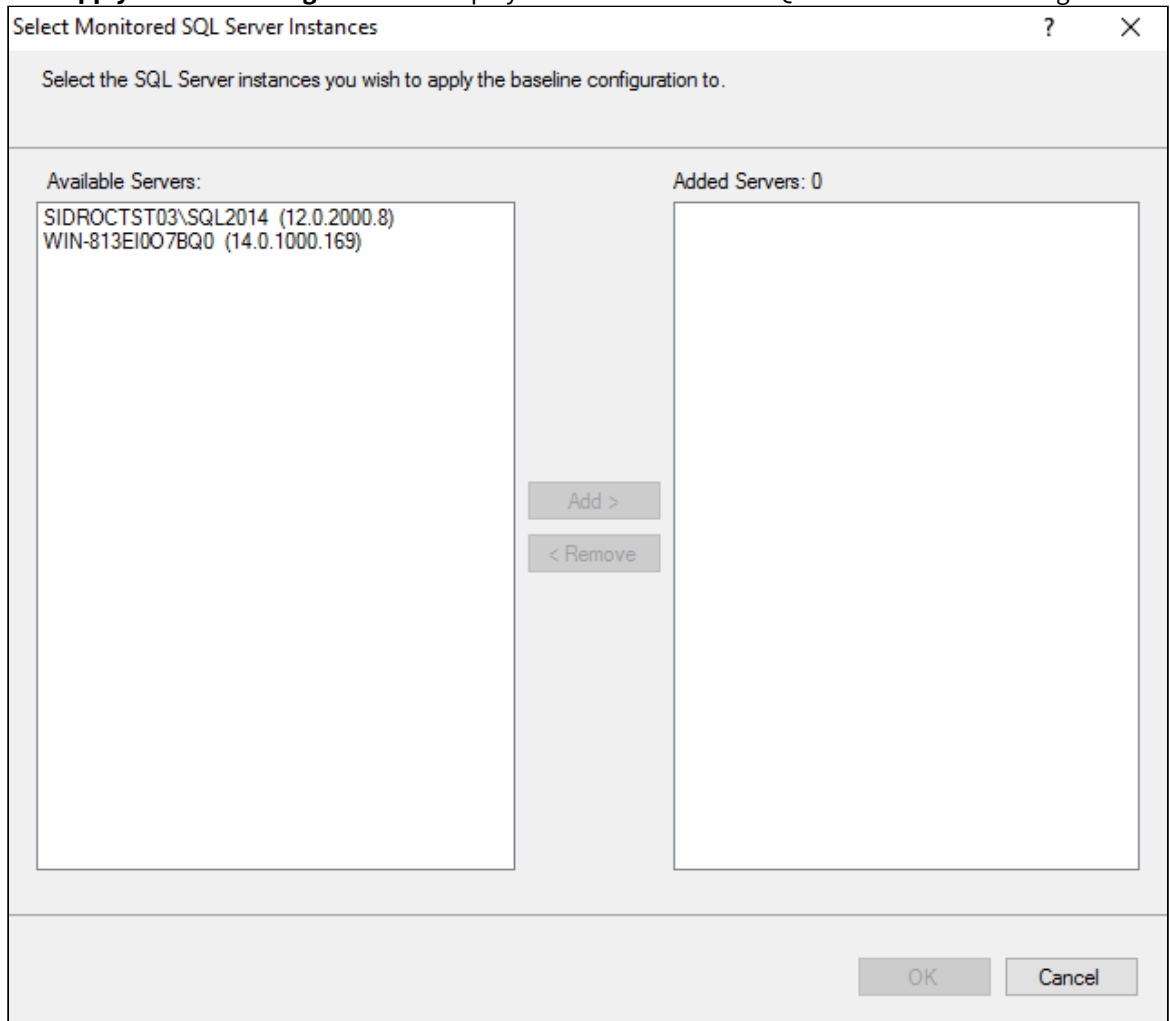
SQL Diagnostic Manager allows you to add and schedule multiple baseline periods in addition to the default one. Open the Add baseline window by clicking the **Manage Baseline** button. For more information, see [Schedule multiple baselines](#).

Apply the baseline to more than one instance

You can apply the baseline configuration from a specific server to additional instances not included when you originally created the baseline configuration.

To add a baseline configuration to additional instances:

1. Click **Apply Baseline Configuration** to display the Select Monitored SQL Server Instances dialog.



2. Select the appropriate instances from the **Available Servers** list, and then click **Add >** to move them to the **Added Servers** list.
3. Click **OK** to confirm your selection and allow SQL Diagnostic Manager to apply this configuration to other instances.

- ✓ When verifying your baseline configuration changes for more than one instance in the [Change Log view](#), note that instance property changes are logged only once. In comparison, baseline configuration changes are logged as many times as the number of instances affected by the change.

Metrics used in baseline calculations

SQL Diagnostic Manager uses the following metrics when calculating a baseline for a monitored SQL Server instance:

- Available Memory
- [Batches Per Second](#)
- [Blocked Sessions](#)
- [Buffer Cache Hit Ratio](#)
- Buffer Cache Size
- Checkpoint Writes Per Second

- [Client Computers](#)
- Data File Count
- Data File Space Allocated
- Data File Space Used
- Data File Space Used (Percent)
- Database Count
- [Disk Queue Length](#)
- [Disk Time \(Percent\)](#)
- Distribution Latency
- Host CPU Usage
- Host CPU Usage MHz
- Host Disk Device Latency
- Host Disk Kernel Latency
- Host Disk Queue Latency
- Host Disk Read
- Host Disk Total Latency
- Host Disk Usage
- Host Disk Write
- Host Memory Swap In Rate
- Host Memory Swap Out Rate
- Host Memory Active
- Host Memory Ballooned
- Host Memory Consumed
- Host Memory Granted
- Host Memory Usage
- Host Net Received
- Host Net Transmitted
- Host Net Usage
- Full Scans Per Second
- Lazy Writer Writes Per Second
- Lock Waits Per Second
- Log File Count
- Log File Space Allocated
- Log File Space Used
- Log File Space Used (Percent)
- [Log Flushes Per Second](#)
- Logins Per Second
- Oldest Open Transaction (Minutes)
- Open Transactions
- OS CPU Privileged Activity
- OS CPU User Time
- OS Memory Used (Percent)
- OS Total Processor Activity
- Packet Errors Per Second
- [Packets Received Per Second](#)
- [Packets Sent Per Second](#)
- Page Life Expectancy
- Page Lookups Per Second
- [Page Reads Per Second](#)
- Page Splits Per Second
- [Page Writes Per Second](#)
- Pages Per Second
- Physical Memory

- Procedure Cache Hit Ratio
- Procedure Cache Size
- [Processor Queue Length](#)
- Read Ahead Pages Per Second
- Replication Latency (Seconds)
- Replication Subscribed
- [Response Time](#)
- SQL Compilations Per Second
- SQL CPU Activity
- SQL Memory Allocated
- [SQL Memory Used](#)
- [SQL Recompilations Per Second](#)
- System Processes
- System Processes Consuming CPU
- Table Lock Escalations
- Tempdb GAM Wait Time Milliseconds
- Tempdb PFS Wait Time Milliseconds
- Tempdb SGAM Wait Time Milliseconds
- Tempdb Size
- Tempdb Size Percent
- Undistributed Transactions
- Unsubscribed Transactions
- Used Memory (Percent)
- User Connections (Percent)
- User Processes
- User Processes Consuming CPU
- Version Store Generation
- Version Store Size
- VM CPU Ready
- VM CPU Swap Wait
- VM CPU Usage
- VM CPU Usage MHz
- VM Disk Read
- VM Disk Usage
- VM Disk Write
- VM Memory Active
- VM Memory Ballooned
- VM Memory Consumed
- VM memory Granted
- VM Memory Swap In Wait
- VM Memory Swap Out Rate
- VM Memory Swapped
- VM Memory Usage
- VM Net Received
- VM Net Transmitted
- VM Net Usage
- Workfiles Created Per Second
- Worktables Created Per Second

Apply baseline to multiple instances

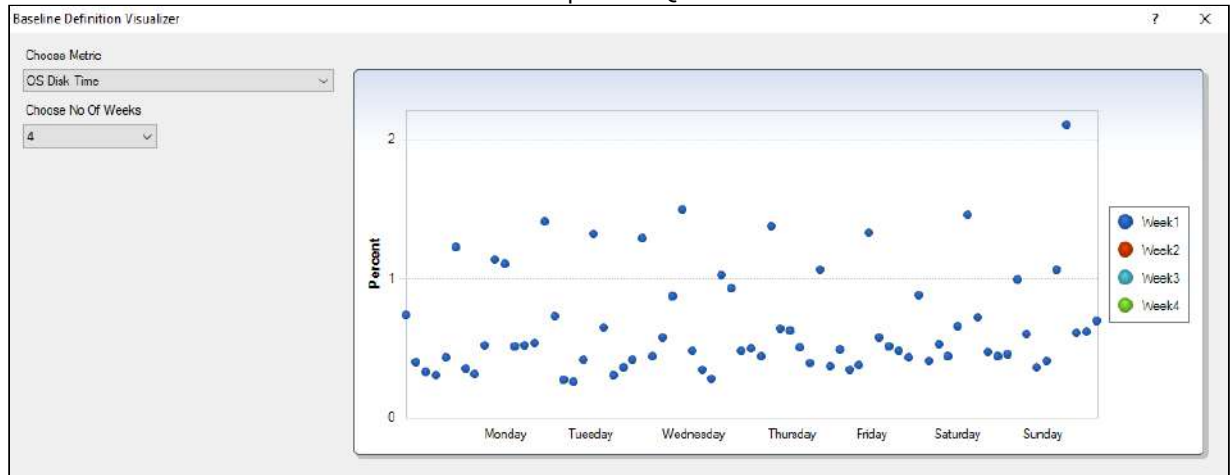
SQL Diagnostic Manager allows you to set a baseline configuration for a monitored SQL Server instance, and then apply that same baseline to other instances. Simply select the appropriate instances from the **Available Servers** list, and click **Add >** to move them to the **Added Servers** list. Click **OK**, and SQL Diagnostic Manager applies this configuration to the other instances.

Use the baseline visualizer

The Baseline Visualizer of SQL Diagnostic Manager provides better visualization of the performance of your monitored SQL Server instance.

To access the Baseline Visualizer:

1. Click **Baseline Visualizer** in the Overview tab of a specific SQL Server instance.



2. In the Baseline Definition Visualizer window, select a metric and the number of weeks you wish data to be returned for the baseline definition.

Schedule multiple baselines

SQL Diagnostic Manager allows you to define and schedule any number of baseline periods in addition to the default one.

Monitored SQL Server Properties - SIDROCTST02\SQL2014

General

Baseline Configuration

Query Monitor

Activity Monitor

Replication

Table Statistics

Custom Counters

Maintenance Mode

OS Metrics

Disk Drives

Cluster Settings

Wait Monitoring

Virtualization

Analysis Configuration

Configure a performance baseline for your monitored SQL Server.

What date range should be used to calculate the performance baseline for this server?

i You should select a date range that represents the typical operation of your monitored SQL Server. This baseline will be used to offer recommendations to help you better configure your alerts in the future.

Dynamic (calculates a dynamic baseline based on the last 7 days of collected data)

Custom

From: 12/25/2018 To: 1/24/2019 **Manage Baseline**

Which days and time period within those days should be used to calculate the baseline?

Baseline Name: Default

Sun Mon Tue Wed Thu Fri Sat

From: 08:00 AM To: 05:00 PM

Would you like to apply this baseline configuration to other SQL Servers?

Apply Baseline Configuration

OK Cancel

To add a new baseline:


1. Click **Manage Baseline** to display the Manage Baseline window.

The screenshot shows the 'Manage Baseline' dialog box with the following elements:

- 2**: A dropdown menu for 'Baseline Name' with the text 'Choose appropriate option from the dropdown'.
- 3**: A section titled 'What date range, days and time period within those days should be used to calculate the performance baseline for this server?' containing:
 - Dynamic (calculates a dynamic baseline based on the last 7 days of collected data)
 - Custom
 - From: 8/13/2008 To: 8/13/2008
 - Sun Mon Tue Wed Thu Fri Sat
 - From: 08:00 AM To: 05:00 PM
- 4**: A section titled 'What date range should be used to apply baseline for this server?' containing:
 - Sun Mon Tue Wed Thu Fri Sat
 - From: 08:00 AM To: 05:00 PM
- 5**: Buttons at the bottom: 'Add', 'Delete', 'Save', and 'Cancel'.


2. Type a name in the **Baseline Name** field.
3. Select **Dynamic** to use the default options or select **Custom** to set the date range, the days, and the time period to calculate the baseline performance for the server.
4. Select the days and time period to schedule when to apply baseline to the server.
5. Click **Add** to save the created baseline, click **Edit** to edit the baseline, or click **Delete**.
6. Click **Save**.

✓ The baseline period you specify should represent a typical operation of the monitored SQL Server instances to which this baseline applies.

 SQL Diagnostic Manager allows only one baseline in effect at any time.

To edit a baseline:

1. Click **Manage Baseline** to display the Add baseline window.
2. Select the baseline you want to update from the **Baseline Name** list.
3. Modify days and time period as desired.
4. Click **Save**.

 When you schedule a new baseline, the changes are applied to the next day or when the baseline is refreshed (test this behavior when the feature is available).

Enable baseline thresholds

SQL Diagnostic Manager allows users to enable alerts when the following metrics are outside their established baseline thresholds:

- Availability Group Estimated Data Loss (Seconds)
- Availability Group Estimated Recovery Time (Seconds)
- Availability Group Log Send Queue Size (KB)
- Availability Group Redo Queue Size (KB)
- Availability Group Redo Rate (KB/sec)
- Availability Group Synchronization Performance (Seconds)
- SQL Server Data Used (Percent)
- SQL Server Log Used (Percent)
- OS Average Disk Queue Length (Count)
- OS Disk Time (Percent)
- OS Memory Usage (Percent)
- OS Paging (Per Second)
- OS Privileged Time (Percent)
- OS Processor Queue Length (Count)
- OS Processor Time (Percent)
- OS User Time (Percent)
- Page Life Expectancy
- Procedure Cache Hit Ratio
- SQL Server CPU Usage (Percent)
- SQL Server Memory Usage (Percent)
- Non-Distributed Transactions (Count)
- Unsubscribed Transactions (Count)
- Unsubscribed Transactions (Seconds)
- Blocked Sessions (Count)
- Client Computers (Count)
- Oldest Open Transaction (Minutes)
- SQL Server Response Time (Milliseconds)
- User Connections (Percent)
- VM CPU Usage (Percent)

Baseline display

SQL Diagnostic Manager allows users to view the baseline in effect in various metric graphs of the Resources tab.

View baseline in effect in the following graphs:

In the **Resources > CPU** view:

- CPU Usage
- Processor Queue Length
- Processor Time

In the **Resources > Memory** view:


- SQL Memory Usage
- Paging
- Cache Hit Ratios
- Page Life Expectancy (sec)

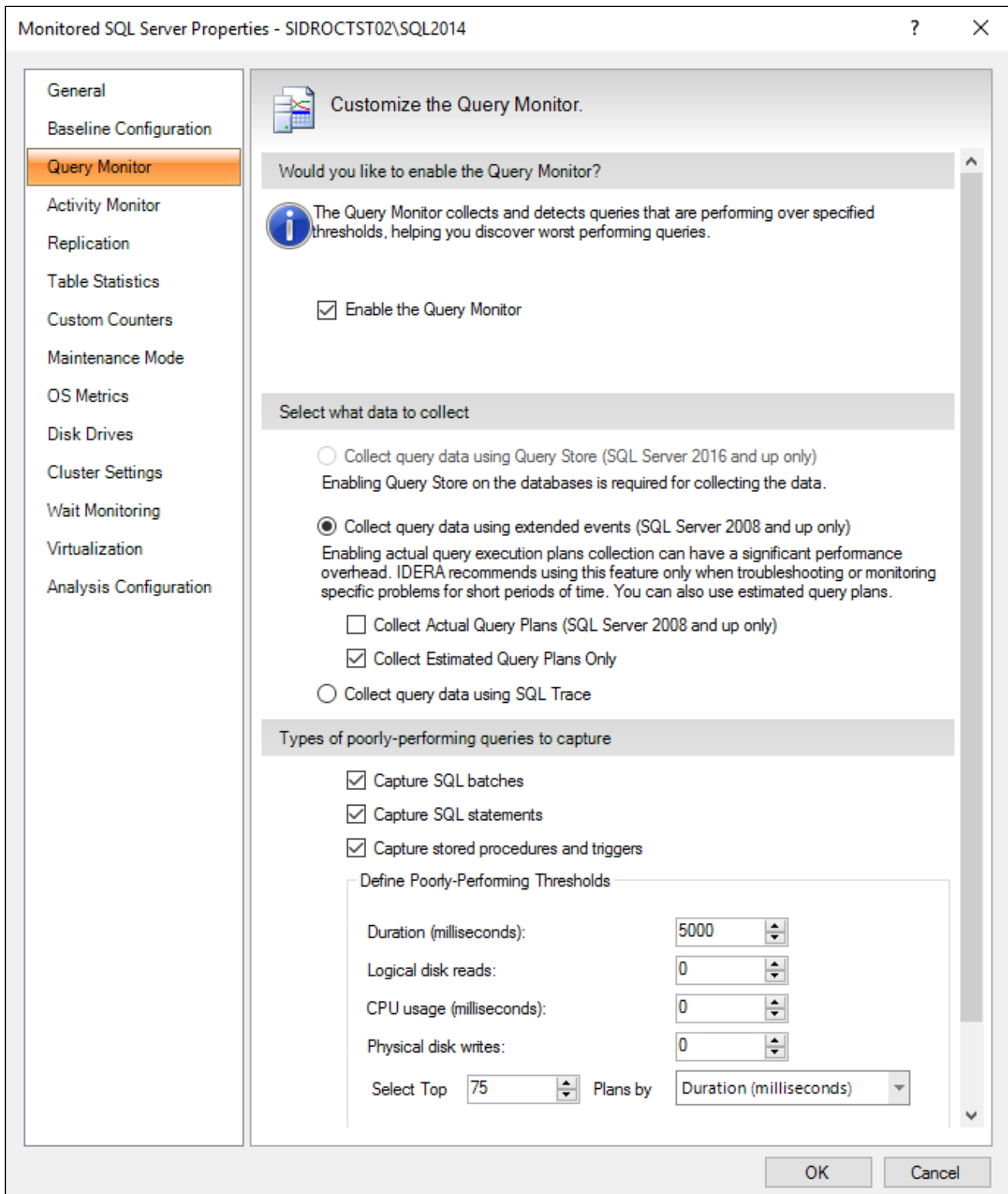
In the **Resources > Disk** view:

- Disk Busy Per Disk
- Average Disk Queue Length Total

Set query monitor options

The Query Monitor is a standard SQL Server trace that collects all the events that occur on your SQL Server instance over a period of time. You can enable this option if you experience query timeouts or other performance issues.

 By default, the Query Monitor option is not enabled.



The Query Monitor window allows you to enable or disable query monitoring using the **Enable the Query Monitor** check box. You can also select the settings that are used:

- [Query data using Query Store](#)
- [Query data using extended events](#)

- [Capturing poorly-performing queries](#)

Query data using Query Store

Collect query data using Query Store for instances running SQL Server 2016 or higher.

- ✔ SQL Diagnostic Manager does not change Query Store selections at this time. Try the [IDERA Query Store Optimizer](#) for more on Query Store improvements.

Query data using extended events

You can select to collect query data using Extended Events (Collect query data using extended events radio button) or SQL Trace (Collect query data using SQL Trace radio button). For instances running SQL Server 2008 or higher, the Query Monitor uses by default Extended Events. First introduced in SQL Server 2008, Extended Events provide a new mechanism to capture information about events inside the Database Engine and diagnose performance problems. This functionality is highly efficient and lightweight. For more information about using Extended Events, see the Microsoft document, [Extended Events](#).

- ⚠ To capture query data for monitored pre-SQL Server 2008 instances, enable the SQL Trace collection option. Enabling this option can degrade performance on your SQL Server.

To collect query execution plans, click **Collect query data using extended events (SQL Server 2008 and up only)** in the Query Monitor window. By default, this option enables the collection of estimated query plans only. If you would like to collect actual query plans instead, select the **Collect Actual Query Plans (SQL Server 2008 and up only)** check box.

- ⚠ Enabling actual query execution plans collection can have a significant performance overhead, so IDERA recommends using this feature only when troubleshooting or monitoring specific problems for short periods of time. You can also use estimated query plans.

Capturing poorly-performing queries

The types of poorly-performing queries to capture include:

- SQL batches
- SQL Statements
- Stored procedures and triggers

You can also define specific poorly-performing thresholds such as duration, logical disk reads, CPU usage, physical disk writes, and select up to 1000 plans defined by these thresholds.

Diagnosing specific performance issues

Select the events that help you best diagnose the specific issues occurring with query performance on your SQL Server instance. For example, you may want to monitor queries taking a long time to complete, requiring heavy CPU usage, and causing a large number of logical disk reads or physical disk writes, which indicates a memory issue.


SQL Diagnostic Manager uses tracing to locate and flag worst-performing T-SQL. SQL Diagnostic Manager cannot display binary data in a readable format. If you captured DTS packages, you may see unreadable characters in the Command field of the Tree View. Capturing DTS data may also adversely affect the performance of the SQL Server or Data Warehouse you are monitoring.

Access the Query Monitor tab

You can access the Query Monitor tab of the Monitored SQL Server Properties window by right-clicking the appropriate SQL Server instance, and then selecting **Properties**. Click **Query Monitor** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.

Configure advanced query monitor options

The Advanced Query Monitor Configuration window allows you to supply filtering criteria such as including or excluding specific applications, databases, and SQL text from the Query Monitor Trace collection. Filtering allows you to improve the results you see in the Query Monitor view and reduce the impact of the Query Monitor trace on your server.

 You can also configure these options directly from the new advanced query views in the SQL Diagnostic Manager web console. See [View your SQL Server queries information](#).

Access the Advanced Query Monitor Configuration window

You can access the Advanced Query Monitor Configuration window (click image to view full size) by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**. Click **Query Monitor** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window, and then click **Advanced**.

Advanced Query Monitor Configuration

Filters

Supply additional Query Monitor filtering criteria.

Which applications would you like to include or exclude?

Exclude: [SQL Server Profiler%]; [SQLAgent%]; [SQLDMO%]; [SQLProfiler]

Include:

use semicolons to separate names; use % for wildcards

Which databases would you like to include or exclude?

Exclude:

Include:

use semicolons to separate names; use % for wildcards

Which SQL text would you like to include or exclude?

Exclude:

Include:

use semicolons to separate names; use % for wildcards

Include SQL Diagnostic Manager queries in the results?

Exclude SQL Diagnostic Manager queries

OK Cancel

- ✓ When using Query Monitor filtering, take into account that Exclude takes precedence over Include.

Using Query Monitor exclusion filtering

Excluding data from your Query Monitor trace helps filter out information that is not important for the task at hand. SQL Diagnostic Manager allows you to filter specific applications, databases, or SQL text and also exclude the SQL Diagnostic Manager queries from the results. When adding multiple items to exclude, use a semi-colon (;) between items of the same type. You can also use a percent character (%) as a wildcard.

To exclude data from your Query Monitor trace while in the Queries view:

1. On the Queries view for the server you want to exclude data from, press the **Configure Query Monitor** button.
2. Click **Advanced**.
3. Enter your filter information in the **Exclude** fields, using (;) to separate items and (%) as a wildcard.
4. Click **OK**.

Using Query Monitor inclusion filtering

SQL Diagnostic Manager allow you to set Query Monitor inclusion filters. You can choose the applications, databases, and SQL text you want to include in the Query Monitor trace and SQL Diagnostic Manager stores only queries that match these inclusion filtering criteria. When adding multiple items to include, use a semi-colon (;) between items of the same type. You can also use a percent character (%) as a wildcard.

To include data in your Query Monitor trace while in the Queries view:

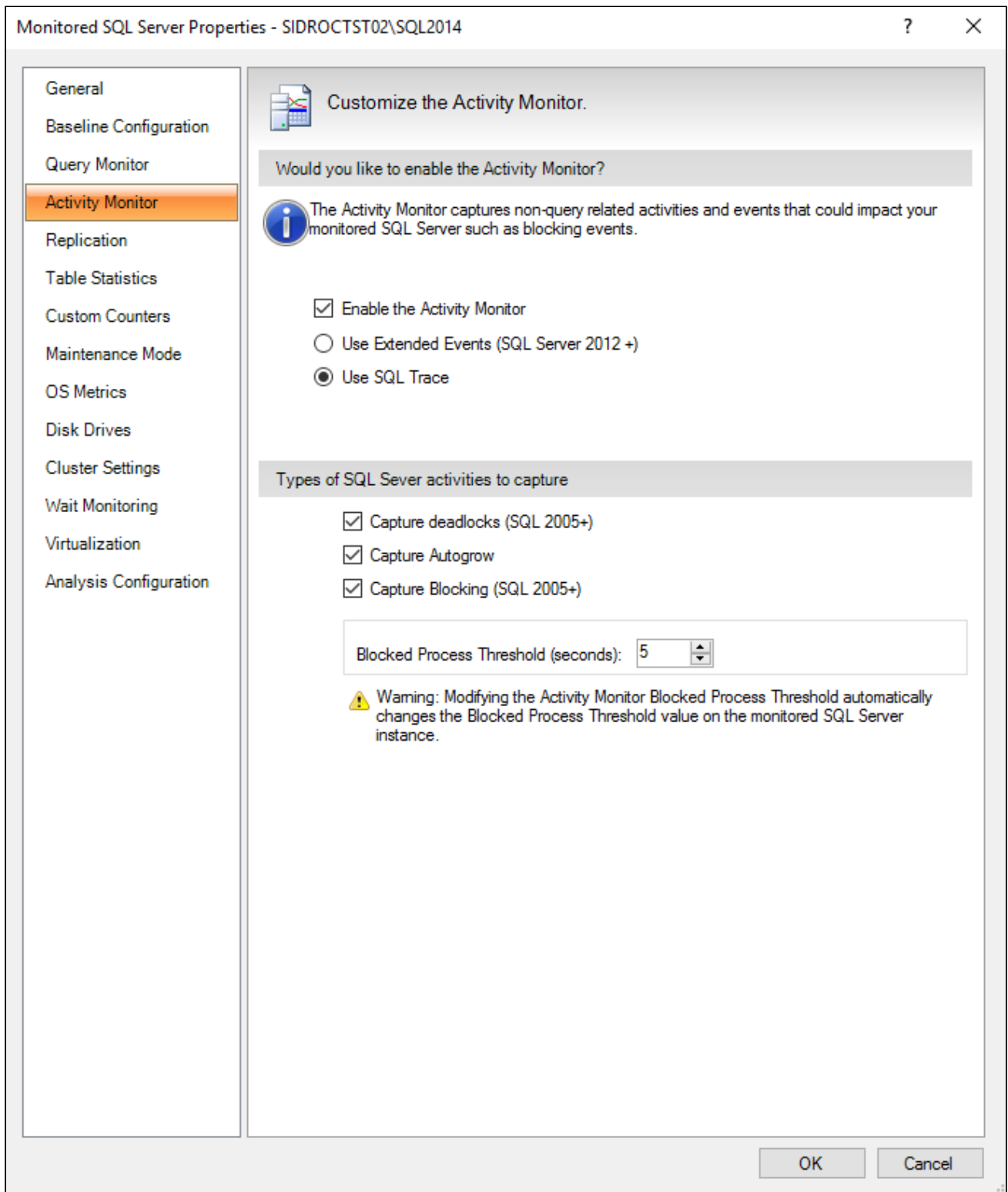
1. On the Queries view for the server you want to include data in, press the **Configure Query Monitor** button.
2. Click **Advanced**.
3. Enter your filter information in the **Include** fields, using (;) to separate items and (%) as a wildcard.
4. Click **OK**.

Set activity monitor options

The Activity Monitor window allows you to enable or disable monitoring of non-query activities. If you enable the Activity Monitor, you can select the settings that are used.


The types of non-query events to capture include:

- [Capture Deadlocks](#) (SQL 2005+)
- [Capture Autogrow](#)
- [Capture Blocking](#) (SQL 2005+)



Non-query activities data collection

You can select to collect non-query activities data using Extended Events (Use Extended Events radio button) or SQL Trace (Use SQL Trace radio button). For instances running SQL Server 2012 or higher, the Activity Monitor uses by default Extended Events. First introduced in SQL Server 2008, Extended Events provide a new mechanism to capture information about events inside the Database Engine and diagnose performance problems. This functionality is highly efficient and lightweight. For more information about using Extended Events, see the Microsoft document, [Extended Events](#).

 To capture non-query activities for monitored pre-SQL Server 2008 instances, enable the SQL Trace collection option. Note that enabling this option can degrade performance on your SQL Server.

Capturing deadlocks

Capturing deadlock information allows the associated deadlock alert to provide details on deadlocked processes on the Alerts view. Deadlock monitoring is supported on monitored instances running SQL Server 2005 or greater and is dependent on the Activity Monitor. To access this information, check **Enable the Activity Monitor** and **Capture deadlocks (SQL 2005+)** in the Activity Monitor window. Details provided include:

- SPID
- Hostname
- Login Name
- Client Application
- Database
- Status
- Wait Time
- Wait Resource
- Transaction Count
- Input Buffer
- Lock Object
- Lock Mode
- Lock Owners
- Lock Waiters

Capturing autogrow events

Capturing autogrow information allows the associated autogrow alert to provide information about log and data growth on the Alerts view. To access this information, check **Enable the Activity Monitor** and **Capture Autogrow** in the Activity Monitor window.

Capturing blocking events

Capturing blocking information allows the associated blocked alert to provide details on blocking and blocked sessions on the Alerts view. Blocking monitoring is supported on monitored instances running SQL Server 2005 or greater and is dependent on Activity Monitor. To access this information, check **Enable the Activity Monitor** and **Capture Blocking (SQL 2005+)** in the Activity Monitor window. Key details provided for blocking and blocked sessions include:

Blocking process details:

- Session ID
- Hostname
- User Name
- Application
- Database

- Last Batch Started
- Transaction ID
- Open Transactions

Blocked process details:

- Session ID
- Hostname
- User Name
- Application
- Database
- Last Batch Started
- Transaction ID
- Open Transactions
- Wait Resource

Blocked process threshold

SQL Server 2005+ includes a new SQL Server Profiler event, the Blocked Process report. This event helps identify blocking issues and provides relevant information to solve these issues. In SQL Diagnostic Manager, you can take advantage of this feature by setting the blocked process threshold value according to your needs. To access the **Blocked Process Threshold** option, enable **Capture Blocking (SQL 2005+)** in the Activity Monitor window. Note that when this option is enabled, the blocked process threshold value is automatically set to 5 (seconds) but can be modified as desired.

The screenshot displays the 'Activity Monitor' configuration window. The left sidebar lists various monitoring options, with 'Activity Monitor' selected. The main panel is titled 'Customize the Activity Monitor' and contains the following elements:

- A question: "Would you like to enable the Activity Monitor?"
- An information icon and text: "The Activity Monitor captures non-query related activities and events that could impact your monitored SQL Server such as blocking events."
- Three radio button options:
 - Enable the Activity Monitor
 - Use Extended Events (SQL Server 2012 +)
 - Use SQL Trace
- A section titled "Types of SQL Server activities to capture" with three checked checkboxes:
 - Capture deadlocks (SQL 2005+)
 - Capture Autogrow
 - Capture Blocking (SQL 2005+)
- A text box for "Blocked Process Threshold (seconds):" with a value of 5, highlighted by a green border.
- A warning icon and text: "Warning: Modifying the Activity Monitor Blocked Process Threshold automatically changes the Blocked Process Threshold value on the monitored SQL Server instance."

Since collecting information about blocked processes in SQL server is resource-intensive, IDERA recommends to set the blocked process threshold value to at least 5 (seconds) or your deadlock monitor runs constantly. For more information on this topic, see the Microsoft document [Increase or Disable Blocked Process Threshold](#).

- ✔ Modifying the Activity Monitor's blocked process threshold automatically changes the blocked process threshold value in your monitored SQL Server instance.

Access the Activity Monitor tab

You can access the Activity Monitor tab of the Monitored SQL Server Properties window by right-clicking the appropriate SQL Server instance, and then selecting **Properties**. Click **Activity Monitor** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.

Disable replication statistics collection

The gathering of replication statistics can require a significant amount of system resources. The Replication tab of the Monitored SQL Server Properties window provides the option to disable the collection of replication statistics and conserve system resources.

Collecting replication statistics requires plenty of system resources, especially when your publisher or distribution queues contain a large number of items. You can easily enable this feature when you need replication statistics, and then disable it once you've collected enough data.

Access the Replication tab

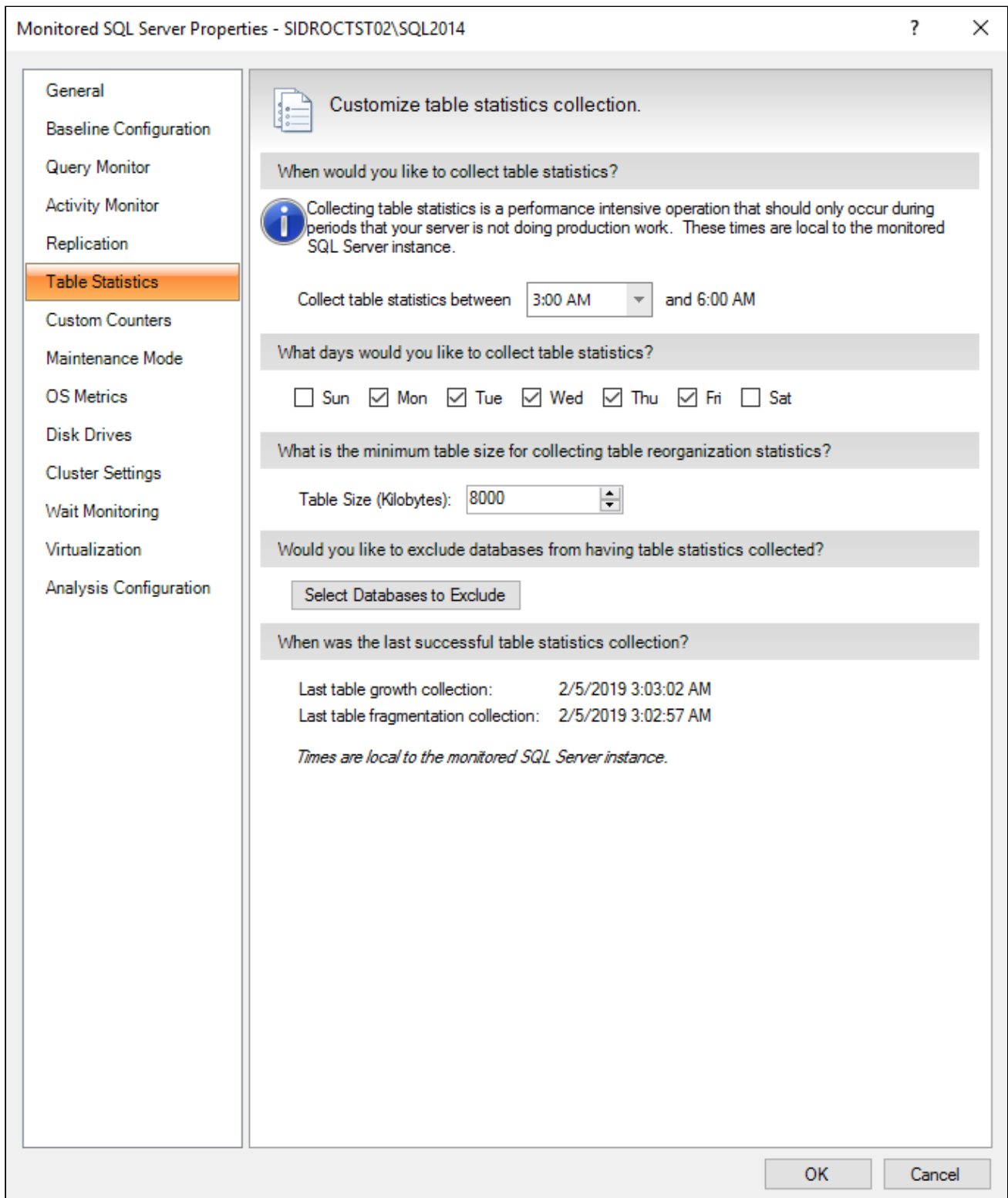
You can access the Replication tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**. Click **Replication** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.

Edit table statistics options

SQL Diagnostic Manager lets you collect statistics and perform trend analysis over time. Table Statistics allow you to designate the days and times SQL Diagnostic Manager collects table growth and reorganization statistics. Because statistics collection can negatively impact performance, SQL Diagnostic Manager allows you to designate an individual Table Statistics collection time for the SQL Server instances. Configure a time you know the instance is utilized less than normal.

In addition, you can limit the amount of performance impact by utilizing the following:

- Limit reorganization to databases over a set size, which eliminates the reorganization of small, rarely-used tables.
- [Exclude non-essential databases from collection](#) by clicking **Select Databases to Exclude** and selecting the appropriate databases.



Access the Table Statistics tab

You can access the Table Statistics tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**. Click **Table Statistics** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.

Schedule table statistics collection

Table Statistics collection is a resource-intensive process. Schedule your collection time for non-peak hours to reduce the possibility of your users experiencing decreased performance.

- ✔ The SQL Diagnostic Manager Collection Service may not include fragmentation data for a particular table for any of the following reasons:
 - Table statistics collection has not occurred.
 - The database hosting the table is excluded from collection.
 - The table size does not meet the minimum size threshold.
 - The database hosting the table is not accessible.
 - The table is locked.

Select a database to exclude from table statistics collection

SQL Diagnostic Manager allows you to exclude non-essential databases from table growth statistics collection. Excluding a database from collection avoid clutter and unnecessary growth in your Repository. To exclude a database, select the appropriate database from the available list, and then click **OK**. You can exclude more than one database at a time.

Access this list of databases for exclusion

If you want to exclude one or more of your databases from table statistics collection, access the Server Properties dialog for that monitored SQL Server instance by right-clicking the instance and selecting **Properties**. Click the Table Statistics tab, and then click **Select Databases to Exclude**.

Link custom counters

The Custom Counters tab allows you to select the metrics you want to explicitly link to the selected SQL Server instance or a group of instances using tags. Note that SQL Diagnostic Manager displays a counter linked through a tag in gray text. You cannot remove this type of counter until you remove the tag from the instance.

Access the Custom Counters tab

You can access the Custom Counters tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**. Click **Custom Counters** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.

Linking a custom counter

To link a custom counter to a selected monitored SQL Server instance, access the Custom Counters tab for that instance. Select one or more counters in the **Available Counters** list, and then click **Add >** to move the counters to the **Linked Counters** list. Use Shift+click or Ctrl+click to select more than one counter. Click **Test** if you want to test the counters. After your test is complete, click **OK**.

Testing a linked custom counter

SQL Diagnostic Manager allows you to test the selected custom counters against your SQL Server instances. When linking the counter, click **Test**. Check the box for all custom counters you want to test, and then click **Test**. SQL Diagnostic Manager indicates that it is running the test, and displays additional information in the **Result** column when the test is complete. Click **Done** to exit the Test Custom Counters window. Be sure to click **OK** after testing the link or SQL Diagnostic Manager does not save your updates.

Schedule maintenance mode

Maintenance mode allows you to temporarily stop alert generation and performance metric collection for the time period that your SQL Server instance is offline. This pause allows you to perform necessary maintenance on a SQL Server without causing SQL Diagnostic Manager to send numerous misleading alerts.

Access the Maintenance Mode tab

You can access the Maintenance Mode function by right-clicking the appropriate monitored SQL Server instance, and then selecting Maintenance Mode. This will prompt three command options: Enable, Disable, and Schedule.

The **Enable** command puts the SQL Server instance into maintenance mode, stopping alert generation and collection of some performance metrics.

The **Disable** command allows SQL Diagnostic Manager to resume performance statistics collection and alert generation for the selected SQL Server instance.

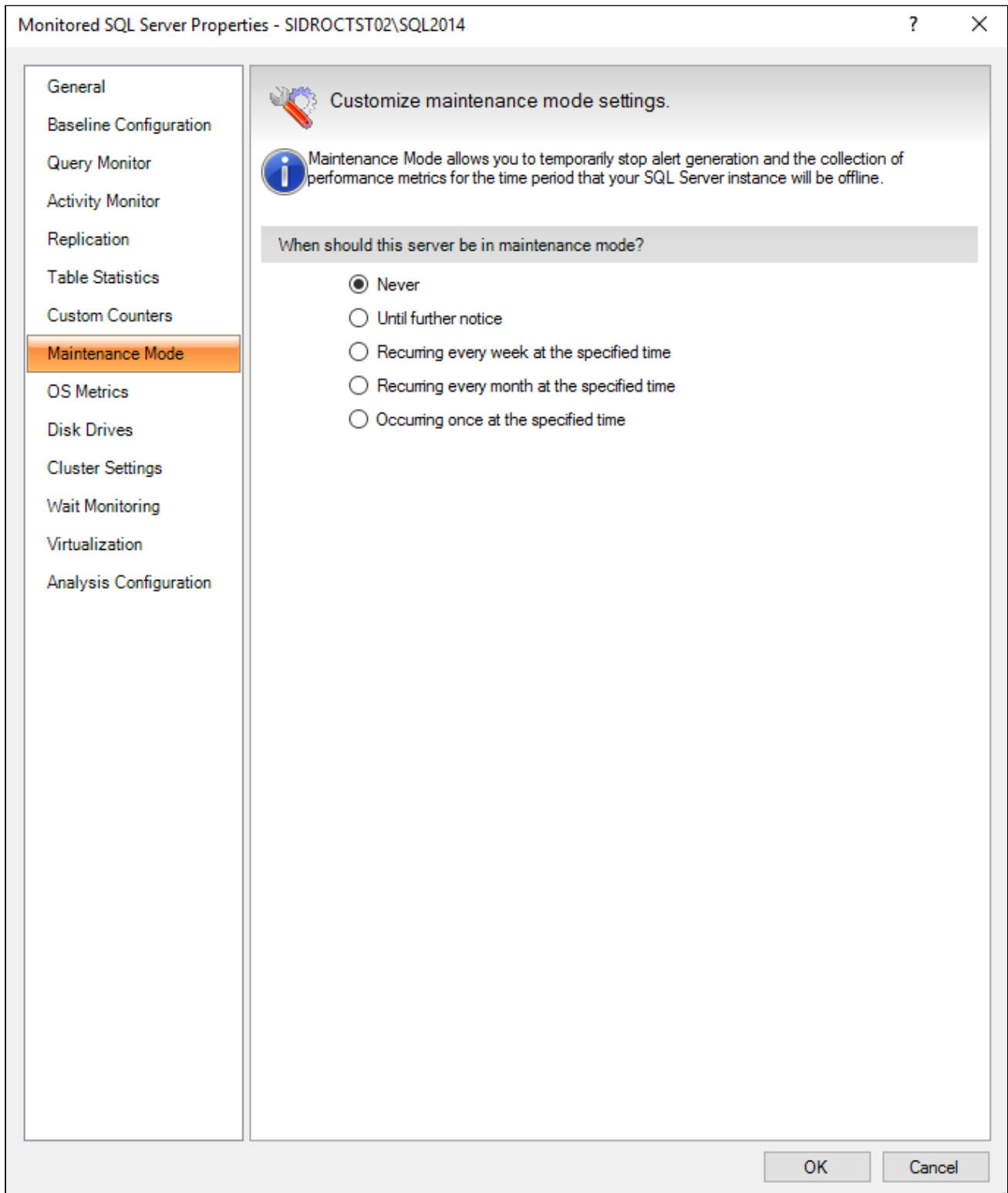
The **Schedule...** command prompts the monitored SQL Server Properties dialog, where under the Maintenance Mode tab it is possible to select a specific maintenance mode setting.

- ✔ A different way to access the Maintenance Mode function is by right-clicking the monitored SQL Server instance, and then selecting Properties. Click Maintenance Mode when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.

About mass maintenance mode

Users looking to apply maintenance mode modifications to a large number of servers at once can run the Maintenance Mode command by right-clicking on the following: Server Tree, a specific tag, a particular server view and Tools.

Available maintenance mode settings



You can choose from the following options when setting a monitored SQL Server instance to maintenance mode:

Never

The server is never in maintenance mode and always generates alerts.

Until further notice

The server is in maintenance mode and does not generate alerts until this setting is changed.

Recurring every week at the specified time

Set the server to enter maintenance mode at regular intervals, for the day, time, and duration you select.

Recurring every month at the specified time


Set the server to enter maintenance mode at regular intervals, for the month, time, and duration you select.

Occurring once at the specified time


The server enters maintenance mode at a specified date and time, and leaves maintenance mode at a specified date and time.

Configure OS metrics monitoring

SQL Diagnostic Manager allows you to enable either OLE automation or direct Windows Management Interface (WMI) procedures for monitoring OS metrics.

 If your SQL Server instance includes enabled Lightweight Pooling, SQL Diagnostic Manager cannot collect OS metrics information.

If SQL Diagnostic Manager cannot provide OS metrics, a message displays on the Dashboard and Services Summary views providing more information.

 When SQL Diagnostic Manager cannot collect OS Metrics due to the disabled collection, a bar displays a link to enable OS metric collection procedures for your SQL Server version.

CPU Usage SQL Diagnostic Manager collects metrics that track the performance of the computers that host your monitored SQL Server instances. These metrics are the following:

- Processor Queue Length
- Memory Paging
- Total Memory Usage
- SQL Server Memory Usage
- Disk Queue Length

OS Metrics tab

You can access the OS Metrics tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance and selecting **Properties**. When SQL Diagnostic Manager displays the monitored SQL Server Properties window, click **OS Metrics** option from the left side tab options.


Once the right-side Properties window displays the OS Metrics configuration options, you find the following options of how the OS metrics are being collected:

- **Do not collect Operating System metrics:** OS metrics collection is not required in SQL Diagnostic Manager. However, if you disable this feature, SQL Diagnostic Manager ceases to collect OS metrics and does not raise any previously-associated alerts. You can easily re-enable OS metric collection at any time if you decide that you want to continue collecting these metrics.
- **Collect Operating System metrics using OLE automation:** Change your OS metrics collection to use OLE automation if you want OS and disk statistics, plus the auto discovery of mount points.

Advantages of using OLE automation

The advantages of using OLE automation are the following:

1. You do not require any additional ports to be opened, since the standard SQL communication port is sufficient.
2. SQL Diagnostic Manager uses the SQL Server service account permissions to capture WMI metrics.
3. And, uses SQL Authentication.


 When using OLE automation you must enable this option; also, OLE automation opens up possibilities for a sysadmin to run queries to reach outside SQL, although you must be a sysadmin to run these queries, it may not be the best option for some environments.

- **Collect Operating System metrics using direct WMI:** Select to collect your OS Metrics using WMI.

Advantages of using direct WMI

The advantages of using WMI are the following:

1. You do not require additional SQL Server configuration adjustments.
2. You do not require SQL Server to gather the performance data because the collection service reaches directly to WMI.

 When using direct WMI you must have RPC port 135 opened for WMI request initiation access. You must have permissions configured on each SQL instance to allow for remote access to WMI, using a Windows account. Other high-end ports opened may be required in firewall to allow WMI to communicate. The following article explains how to set a fixed port for WMI: [Setting Up a Fixed Port for WMI](#)

• **Required permissions for direct WMI collection**

It is preferred that the WMI user is a local administrator on the monitored SQL Server instance. However, if you do not want to grant administrator access, use the following steps to configure remote WMI access in Microsoft Windows:

1. Add the user account to the Performance Log Users and distributed COM users groups.
2. Grant privileges to WMI.

You also may need to add the WMI user account to the following policies:

- Act as part of the operating system
- Log on as a batch job
- Log on as a service
- Replace a process level token

For more information about using a direct WMI connection, see the Microsoft document, [Securing a Remote WMI Connection](#).

• **Configure WMI to run out-of-process**

Edit your specific registry settings to allow WMI to run out-of-process. To edit these registry settings, perform the following steps:

1. In the cd image of the SQL Diagnostic Manager install, open Scripts folder.
2. Copy the SQLdmoutofprocoleautomation.reg file onto each of your monitored SQL Server instances.
3. Run this file on each of your monitored SQL Server instances.

• **Configure WMI timeouts**

SQL Diagnostic Manager allows you to configure the WMI timeout value. In some environments, customers may experience WMI timeouts when their machine is too busy to respond in a timely manner. Although WMI timeouts should not be ignored and often point to an environmental issue, you can control this value when the behavior displayed is normal in your environment. The default WMI timeout value in SQL Diagnostic Manager is of 90 seconds to reduce the number of intermittent errors.

To configure the WMI timeout value:

1. Navigate to the SQL Diagnostic Manager install directory
2. Open SQLdmCollectionService.exe.Config
3. Modify the **<CollectionService>** node under the **<Services>** node
4. Add wmiQueryTimeOut="300" as a parameter to this node
5. Save and close file

6. Restart the collection service

Sample configuration entry:

```
<Services>  
<CollectionService instanceName="Default" servicePort="5167"  
managementServiceAddress="Server" managementServicePort="5166"  
heartbeatIntervalSeconds="180" wmiQueryTimeOut="300"/>  
</Services>
```

Issues you may encounter before SQL Diagnostic Manager begins collecting OS Metrics:

The following issues require user action before SQL Diagnostic Manager can begin collecting OS metrics:

- **OLE Automation Procedures Disabled:** Select the **Collect Operating System data using OLE Automation** option to enable OLE Automation procedures.
- **OLE Automation Procedures Unavailable:** The stored procedures associated with OLE Automation are missing on the SQL Server instance. SQL Diagnostic Manager cannot capture OS metrics until these stored procedures are available on the SQL Server instance.
- **WMI Service Not Running:** The WMI Service on the monitored SQL Server instance may not be running. SQL Diagnostic Manager cannot collect OS metrics until this service is started.
- **Lightweight Pooling Enabled:** This is an advanced feature used in symmetric multiprocessing environments in SQL Server. SQL Diagnostic Manager cannot collect OS metrics until this option is disabled on your monitored SQL Server instances.

Customize disk statistics

Disk drive statistics collection provides the ability for SQL Diagnostic Manager to alert on disk drives and database space calculations. By default, SQL Diagnostic Manager attempts to auto-discover local disk drives on monitored instances. In some environments, the auto-discovered list may be incomplete or incorrect. The Customize disk drive statistics collection window allows you to turn off the auto-discover feature and manually add the drives for drive statistics collection.

- ✔ SQL Diagnostic Manager populates the disk drive list using operating system metrics from a WMI call even if your collection is configured to use OLE automation. SQL Diagnostic Manager requires you to enter credentials to connect directly to WMI. This functionality lets SQL Diagnostic Manager quickly discover mount points while the OLE automation does not provide such information.

Access the Disk Drives tab

You can access the Disk Drives tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**. Click **Disk Drives** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.

Incomplete disk drive list

SQL Diagnostic Manager automatically discovers local disk drives on your monitored instances. Sometimes this list is incomplete or contains drives that are not related to SQL Server. If you need to modify this list, clear the **Discover disk drives** check box, and then type the drive information including mount points, if necessary, into the **Available disk drives** field. Select the appropriate drive or drives and then click **Add** to move the drive information to the **Selected disk drives** box. Click **OK**.

Configure cluster settings

If you have a clustered environment, SQL Diagnostic Manager can trigger an alert whenever a cluster is running on a non-preferred node. The preferred cluster node is the "active" node in a cluster. By default, the preferred cluster node is the first node added when deploying SQL Diagnostic Manager in your clustered environment.

Access the Cluster Settings tab


You can access the Cluster Settings tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**. Click **Cluster Settings** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.

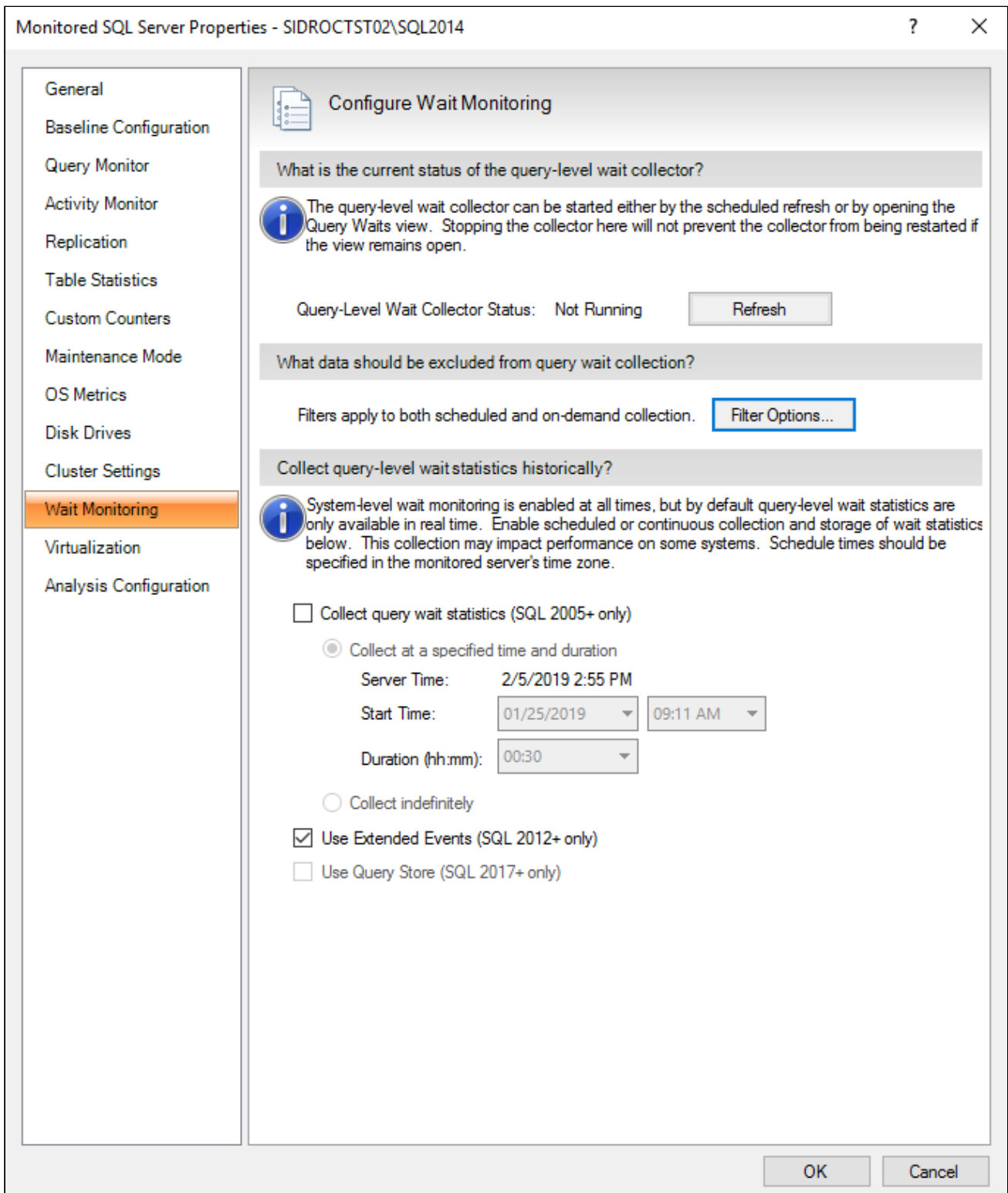
Configure wait monitoring

SQL Diagnostic Manager allows you to collect and view wait statistics on the selected SQL Server instance. Wait statistics offer one of the best means in diagnosing the reason a bottleneck is occurring on your system.

Query Store


Query Store is now an option for query plan collection and monitoring. Query Store must be enabled on the monitored instance, along with Query Monitor and Query Waits.

 SQL Diagnostic Manager does not change Query Store selections at this time. Try the [IDERA SQL Query Store Optimizer](#) for more on Query Store improvements



The Wait Monitoring tab of the Monitored SQL Server Properties window allows you to configure how SQL Diagnostic Manager collects wait statistics.

- View, start, stop, and refresh the query-level wait collector.
- Collect query-level wait statistics historically and set the time to collect them.

 Overly complex filters of SQL Text using Extended Events can cause high loads on the monitored instance due to a known Microsoft defect.

When to use wait monitoring

Wait Monitoring helps you diagnose congestion or blocks on your system. You should enable Wait Monitoring for the time period required to diagnose specific issues you experience. When your diagnosis is complete, disable Wait Monitoring.

Access the Wait Monitoring tab

You can access the Wait Monitoring tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**. Click **Wait Monitoring** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.

Configure advanced query wait statistics options

The Advanced Query Wait Statistics Configuration window allows you to filter out applications, databases, SQL text from the Query Waits Statistics collection, and exclude specific SQL Diagnostic Manager queries as well as limit the number of rows to be collected at a time. Filtering allows you to improve the results you see in the Query Waits view and reduce the impact of the Query Wait trace on your server.

⚠ Overly complex filters of SQL Text using Extended Events can cause high loads on the monitored instance due to a known Microsoft defect.

Access the Advanced Query Wait Statistics Configuration window

You can access the Advanced Query Wait Statistics Configuration window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**. Click **Wait Monitoring** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window, and then click **Filter Options**.

✔ When using Query Wait filtering, take into account that Exclude takes precedence over Include.

Using Query Wait filtering

Excluding data from your Query Waits Statistics collection helps filter out information that is not important for the task at hand. SQL Diagnostic Manager allows you to filter specific applications, databases, SQL text and also exclude the SQL Diagnostic Manager queries from the results or limit the number of rows to be collected at a time. When including multiple items to exclude, use a semi-colon (;) between items of the same type. You can also use a percent character (%) as a wildcard.

To exclude data from your Query Wait Statistics collection while in the Queries view:

1. On the Queries Waits view for the server you want to exclude data from, press the **Configure** button.
2. Click **Filter Options**.
3. Enter your filter information in the provided fields, using (;) to separate items and (%) as a wildcard.
4. Click **OK**.

Configure virtualization

The Virtualization tab of the Monitored SQL Server Properties window displays the name of the host server on which the selected monitored SQL Server instance is running. If this is not the correct host server, click **VM Configuration** to make the necessary changes to [connect to the appropriate virtual machine](#).

Access the Virtualization tab

You can access the Virtualization tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**. Click **Virtualization** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.

11.0 Copy Dashboard for Azure SQL

The Dashboard view contains a graphical representation of the activity on your SQL Server instance. This view not only displays the status of your SQL Server instance, but it allows you to diagnose where problems originate.

The Dashboard view is made up of panels, which include additional information for specific metrics in [chart form](#). Hover your mouse cursor over an area within a chart on which you want to get actual statistics. Click a specific metric to drill down for more detailed information. By using the options in the menu available when you right-click over an area, you can configure alerts, view associated help topics and details, print, save the image, or export the dashboard to Microsoft Excel as a .csv file.

You can also customize your Dashboard view to show only those panels that are important to your monitored SQL Server instance. SQL Diagnostic Manager also allows you to save your customized view on a per-instance basis or use the same view for all of your instances. For more information about customizing your Dashboard view, see [Customize your Dashboard view](#).

Access the Dashboard view

You can open the Dashboard view of the SQL Diagnostic Manager Overview tab by selecting the appropriate SQL Server instance, and then clicking **Overview > Dashboard**.

Dashboard view

The Dashboard view is organized with panels that display all the key metrics collected by SQL Diagnostic Manager by category.

The available data panels for the Dashboard view include:

Cache

The [Cache panel](#) tracks the database and procedure processes using a buffer pool of memory on your monitored SQL Server instance, and includes the following charts and gauges:

- Page Life Expectancy
- Areas
- Hit Ratios

CPU

The [CPU panel](#) includes key CPU metrics for your SQL Server instance and the computer that hosts the instance. The metrics displayed in CPU include the following charts and gauges:

- Usage
- Processor Queue Length
- Call Rates

Custom Counters

The [Custom Counters panel](#) tracks selected custom counters for your monitored SQL Server instance. You must have at least one [custom counter](#) to view data in this panel.

Databases

The [Databases panel](#) includes key metrics covering the health of the databases on your monitored SQL Server instance. The metrics displayed in Databases include the following charts and gauges:

- Transactions/sec

- Log Flushes/sec
- Reads/sec
- Writes/sec
- I/O Stall ms/sec

Disk

The [Disk panel](#) includes key metrics that monitor the health of the disk of your monitored SQL Server instance. The metrics displayed in Disk include the following charts and gauges:

- Latency
- Throughput
- SQL Server Physical I/O

File Activity

The [File Activity panel](#) includes the top five database files with the highest relative activity since the last refresh of your monitored SQL Server instance. You can select different metrics and display results for all files, or a single database, disk, or file. The available metrics include:

- Reads/sec
- Writes/sec
- Transfers/sec

Lock Waits

The [Lock Waits panel](#) tracks the total number waits caused when a task is waiting to acquire a lock on your monitored SQL Server.

Memory

The [Memory panel](#) includes key Memory metrics for your SQL Server instance and the computer that hosts the instance. The metrics displayed in Memory include the following charts:

- SQL Server Usage
- SQL Server Paging
- VM Memory Usage
- Host Memory Usage

Network

The [Network panel](#) tracks the performance of the network connection being used by your monitored SQL Server, and includes the following charts and gauge:

- SQL Server Throughput
- VM Network Usage Throughput
- Host Network Usage Throughput
- Response Time

Server Waits

The [Server Waits panel](#) tracks the total number and time spent on waits affecting your monitored SQL Server.

Sessions

The [Sessions panel](#) includes session-specific data items. The metrics displayed in Sessions include the following chart and gauge:

- Activity & Blocking
- Clients

Tempdb

The [Tempdb panel](#) tracks the status of the tempdb database on your monitored SQL Server instance, and includes the following charts and gauge:

- Tempdb Space Used
- Tempdb Contention
- Version Store Cleanup Rate

Virtualization

The [Virtualization panel](#) tracks the performance of the host server that manages your virtual machines and the VM on which the monitored SQL Server instance is running, and includes the following charts and gauge:

- Virtual Machine Memory Areas
- VM Host Memory Areas
- Virtual Machine Disk Usage
- VM Host Disk Usage
- CPU Ready

Understand the Dashboard colors

The following colors are associated with a status for each of the provided panels.

Gray

Indicates the status of the data is normal, or within the thresholds assigned to the metric.

Blue

Indicates the status of the data is at an informational level based on your settings.

Yellow

Indicates the status of the data is at a warning level, or outside the threshold of normal but not yet at the critical status.

Red

Indicates the status of the data is at a critical level and requires immediate action.

Blank metrics on the Dashboard

If a metric on the Dashboard has no value, or displays as blank, SQL Diagnostic Manager is not able to collect the corresponding counter. Causes of this situation include:

- The SQL Diagnostic Manager Console has not captured the metrics. It can take up to two refreshes for this capture to occur.
- OS metrics are disabled. SQL Diagnostic Manager requires direct WMI or OLE automation to collect data.
- If using OLE automation, those procedures are unavailable. SQL Diagnostic Manager is unable to locate required OLE automation stored procedures.
- If connecting to WMI directly, the WMI Service is not running. SQL Diagnostic Manager requires that the WMI service is running on the computer that hosts the SQL Server instance in order to collect OS metrics.

- The counters do not exist on this SQL Server instance.

You can [enable OS metrics monitoring](#) to track the performance of the computers on which your monitored SQL Server instances are hosted.

Refresh your Dashboard data

The Dashboard view is refreshed according to the time set in the **Server View Refresh** field on the Console Options window, which is accessible by clicking **Tools > Console Options**.

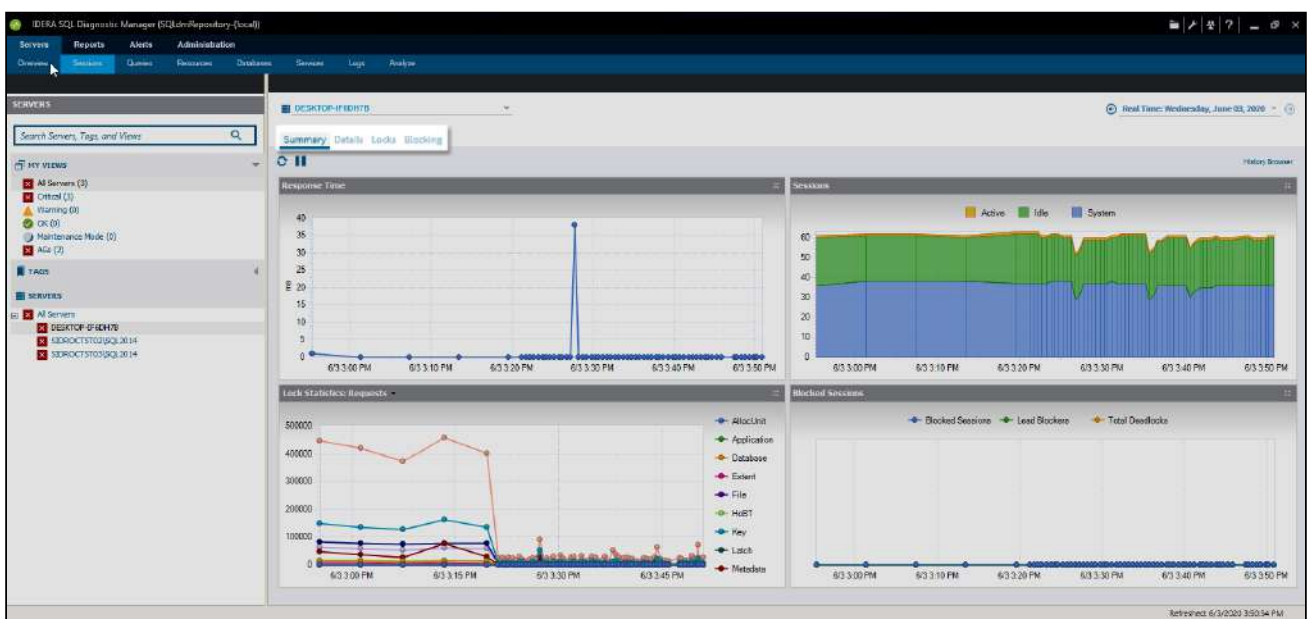
Understanding your chart data and refresh interval

SQL Diagnostic Manager charts provide data whether collected real-time or as part of a scheduled collection event. For additional information about how SQL Diagnostic Manager displays your chart data, see [Charts](#).

Monitor sessions

The Sessions tab contains the following views:

- **Summary:** You can view charts for each Sessions statistic, the following charts display in the Summary tab:
 - Response Time
 - Sessions
 - Lock Statistics: Average Wait Time, Deadlocks, Requests, Timeouts, Waits, and Wait Time
 - Blocked Sessions
- **Details:** Allows you to create charts with the specific data items you need.
- **Locks:** Allows you to create charts based on each lock present on your SQL Server.
- **Blocking:** Allows you to create charts specific to all the Blocking sessions.



The Sessions tab provides the following options:

- The ability to diagnose performance bottlenecks caused by blocking or locked sessions.
- The ability to trace specific sessions to see why the session may be blocking or locked.
- The ability to kill sessions to improve system performance.

Access the Sessions tab

Access the Sessions tab by clicking the appropriate monitored SQL Server instance and clicking the Sessions tab.

Alerts associated with the Sessions tab

The following alerts are associated with the Sessions tab:

Blocked Sessions (Count)

The number of sessions blocked by other sessions holding requested locks.

Blocking Session Wait Time (Seconds)

The amount of time (in seconds) that a SQL Server session is blocking other sessions.

Client Computers (Count)

The number of unique client computers connected to the instance.

Deadlock

Refers when two processes are waiting for a resource held by the other process.

Oldest Open Transaction (Minutes)

Displays the longest amount of time between a transaction using a Begin Transaction statement and the corresponding Commit Transaction or Rollback Transaction statement, in minutes.

Query Monitor Events (Count)

The number of queries captured by Query Monitor on the monitored SQL Server instance during the last scheduled refresh.

Session CPU Time (Seconds)

Provides the CPU time (in seconds) used by a SQL Server sessions.

SQL Server Response Time (Milliseconds)

Calculates the sum of the response time (in milliseconds) to send a batch of SQL statements to the server, process the query, and return the results to the SQL Diagnostic Manager Console.

User Connections (Percent)

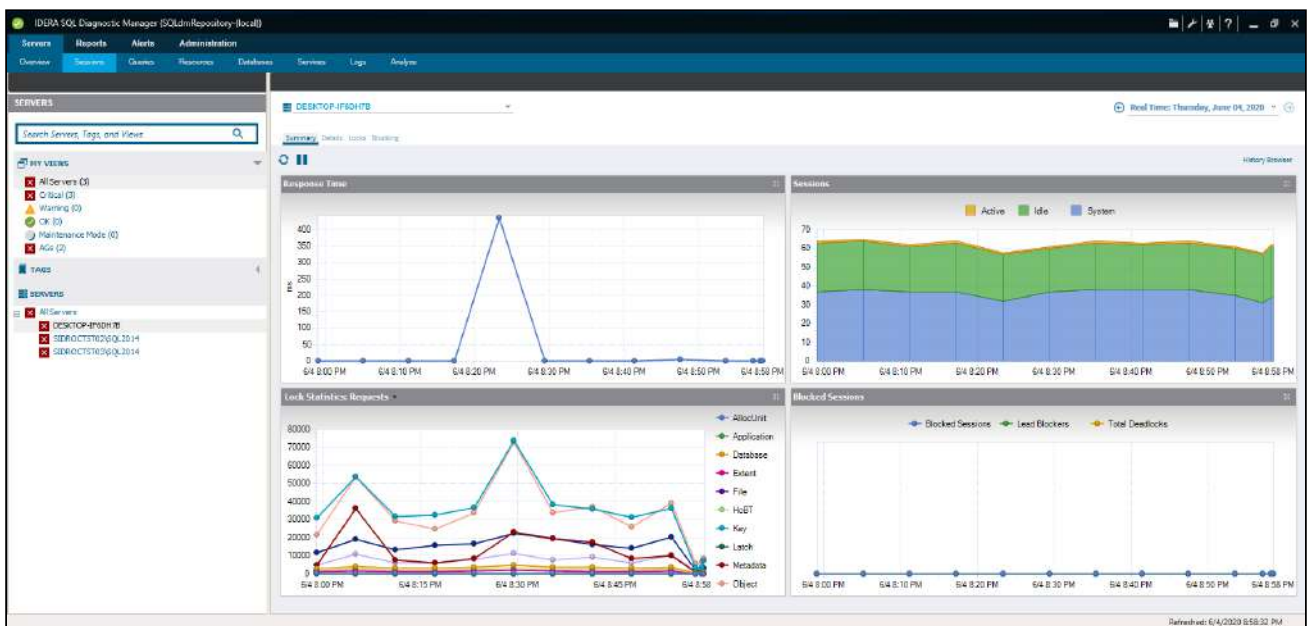
The percentage of user connections allowed that are currently used.

Get sessions performance summary

The Sessions Summary view displays real-time charts that allow you to quickly view key diagnostic statistics for sessions on your SQL Server instance. The Sessions Summary tab includes the Response Time, Sessions, Lock Statistics, and Blocked Sessions charts.

Access the Summary view

You can open the Summary view of the SQLDM Sessions tab by selecting the appropriate SQL Server instance, and clicking **Sessions > Summary**.



- Response Time chart

The Response Time chart displays the response time (in milliseconds), you can see the specific time and response time of any data point in the chart by hovering your mouse over it. The Response Time chart indicates the current speed and congestion of the network as well as the speed with which SQL Server processes small queries. Unusually high response times could indicate the need to upgrade server or network hardware.

- Sessions chart


The Sessions chart displays the number of concurrent connections to the monitored SQL Server instance over time. The total number of sessions includes the Active, Idle, and System sessions. Also, you can see the specific metric value for a given time by hovering your mouse over the chart.

- Lock Statistics chart

The Lock Statistics chart provides a breakdown of the various lock types taken by SQL Server in the last refresh. The Lock Statistics chart offers, by using the drop-down menu, charts that include information on the Average Wait Time, Deadlocks, Requests, Timeouts, Waits, and Wait Time. Also, you can see the specific time and statistic on any data point on the Lock Statistics for a given time by hovering your mouse over the chart.

- Blocked Sessions chart

The Blocked Sessions charts lists all the Blocked Sessions, Lead Blockers, and the Total Deadlocks, in real-time, for the selected SQL Server instance, allowing you to easily see blocks as they occur. Also, you can see the specific time and number of blocked sessions at a given time by hovering your mouse over the chart.

 Right-click any of the Summary Sessions graphs to enable the Toolbar option. This toolbar helps you set different configurations, like copying the graph to clipboard, changing the type of graph, or the color palettes, and printing. Change the view of the graph to 3D/2D, rotated view, rotate around Y-axis, rotate around X-axis, clustered Z-axis, change the axes settings, add point labels, data grid, or a legend box. Also, you can access to other graph Properties.

Get sessions performance details

The Sessions Details view provides an in-depth analysis of sessions running on your monitored SQL Server instance. You can view a wide range of information from performance details to open transactions and configured options or track process activity at the statement level. Individual sessions appear in the top portion of the window.

Right-click any session in the list to View Locks, Show Query History, Trace Session, Kill Session, Print the associated statistics, or Export statistics to Microsoft Excel.

Session ID	Type	Blocked By	Blocking	CPU (ms)	Memory Usage (kb)	Physical I/O	Wait Time (ms)	User	Host	Database	Status	Open Transactions	Command	Application
65			False	4,986	15	82,777	0	IDERA\F00E\VIDERA	SIROCTST03	msdb	Sleeping	0	SQLAgent - Job invocation engine	
75			False	31	16	0	0	NT SERVICE\ReportServer	ReportServerSQL2014	ReportServer	Sleeping	0	Report Server	
58			False	5	0	72	0	S24\NT AUTHORITY\SYSTEM	master	master	Suspended	0	EXECUTE	Microsoft® Windows® Operating System

When you select a session, SQL Diagnostic Manager displays the statistics for that session in the **Details** section at the bottom of the window. This section includes in-depth information regarding the session Connection, Usage, Lock, and Tempdb usage.

Access the Details view

To open the Details view of the SQL Diagnostic Manager Sessions select the the appropriate SQL Server instance, and click **Sessions > Details**.

The screenshot shows the SQL Diagnostic Manager interface. On the left, there is a sidebar with 'SERVERS' and 'TAGS'. The main area displays a list of sessions for 'DESKTOP-IFSDH7B'. The 'Filter' button is highlighted in blue. Below the list, the 'Details' view for session 58 is shown, including connection information, usage statistics, lock information, and tempdb usage.



Select the **Filter** button to select items shown on the view.

Filter views

On a number of views, SQL Diagnostic Manager includes filtering options ranging from ribbons allowing you to select items shown on the view, to a **Filter Settings** window that provides a variety of options to customize how SQL Diagnostic Manager presents your view.

Use the Filter options to customize your view

The Filter box contains the filtering options available for that view, meaning that the available options are not consistent across views because of the data displayed in each view. You can select the options you want displayed in the menu by using the Filter Settings window. Click the **Filter** icon in the ribbon to display the Filter Settings window.

Every Filter Settings window in SQL Diagnostic Manager provides specific options for the current view. These options can include true or false fields that allow you to decide whether to show a particular item, filtering fields where you can enter an item name or even use a wildcard (%), dates and time ranges, and interval periods:

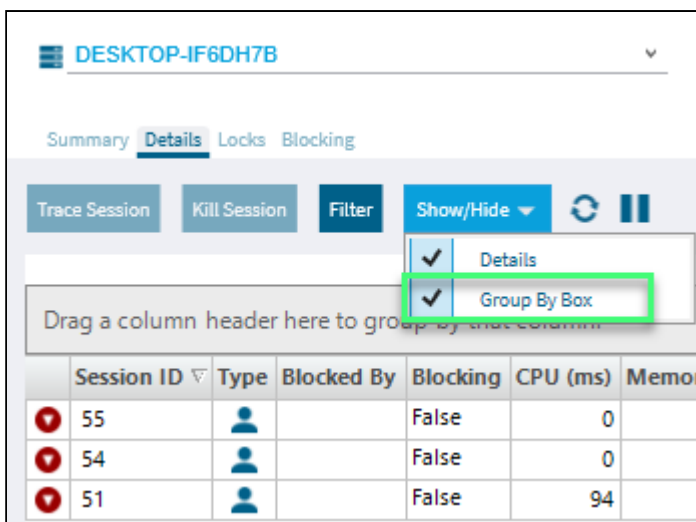
Filter Settings	
Exclude Filters Exclude Application Exclude Host	
Include Filters Include Application Include Host	
Resources	
Active only	False
Affecting tempdb only	False
Blocked only	False
Blocking and Blocked only	False
Blocking or Blocked only	False
Consuming CPU only	False
Exclusive Locks only	False
Holding Locks only	False
Lead Blockers only	False
Open Transactions only	False
System	
Exclude SQLDm	True
User Only	True
Active only Show only Sessions that are running or runnable	
<input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	

- **Exclude Filters**
 - Exclude Application: Specify the Application you want to exclude in the filter, you can enter the application name or use the wildcard character.
 - Exclude Host: Specify the Host name you want to exclude in the filter or use the wildcard character.
- **Include Filters**

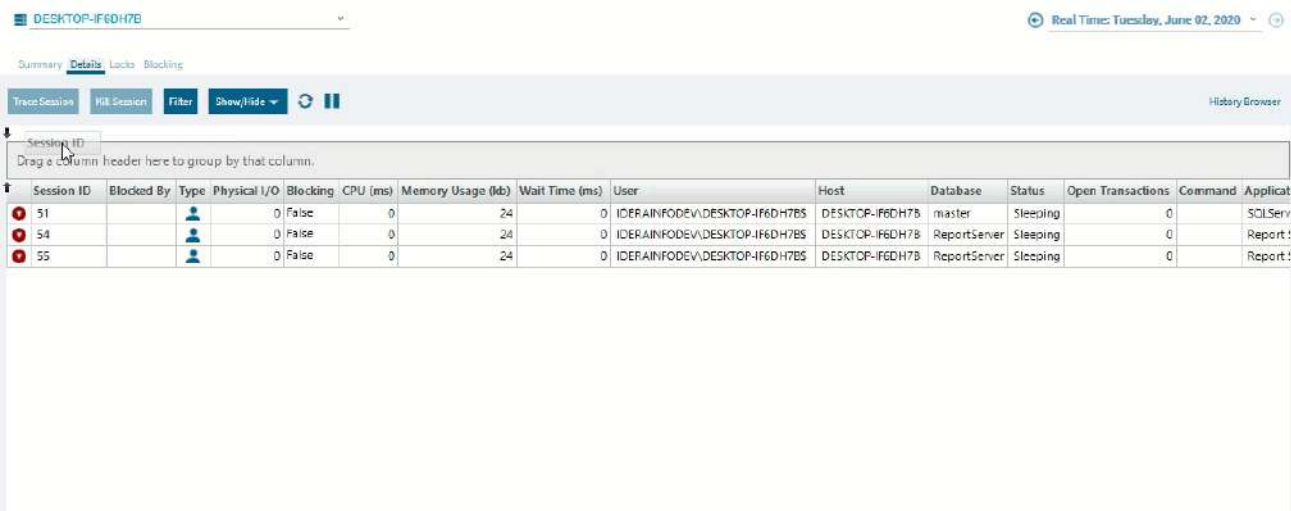
- Include Application: Specify the Application you want to include in the filter, you can enter the application name or use the wildcard character.
- Include Host: Specify the Host name you want to include in the filter or use the wildcard character.
- **Resources**
 - Active only: Select True or False to show only Sessions that are running.
 - Affecting tempdb only: Select True or False to show only Sessions that are affecting tempdb (2005+only)
 - Blocked only: Select True or False to show only Sessions that are blocked by other process.
 - Blocking and Blocked only: Select True or False to show only Sessions that are being blocked by other session and are blocking other sessions.
 - Blocking or Blocked only: Select True or False to show only Sessions that are being blocked by another session or are blocked other sessions.
 - Consuming CPU only: Select True or False to show only Sessions that are consuming CPU cycles.
 - Exclusive Locks only: Select True or False to show only Sessions that holding exclusive locks.
 - Holding Locks only: Select True or False to show only Sessions that are holding locks.
 - Lead Blockers only: Select True or False to show only Sessions that are leading a blocking chain.
 - Open Transactions only: Select True or False to show only Sessions that have open transactions.
- **System**
 - Exclude SQLdm: Select between True or False to exclude all Sessions associated with SQL Diagnostic Manager.
 - User Only: Select between True or False to show only User Sessions and exclude all the System sessions.

Use the Group By feature to organize your view

The Group By feature allows you to group the information shown in your views. To group your information by a column, check the **Group By Box** option, and then drag one or more column headings into the provided area as shown in the following image.



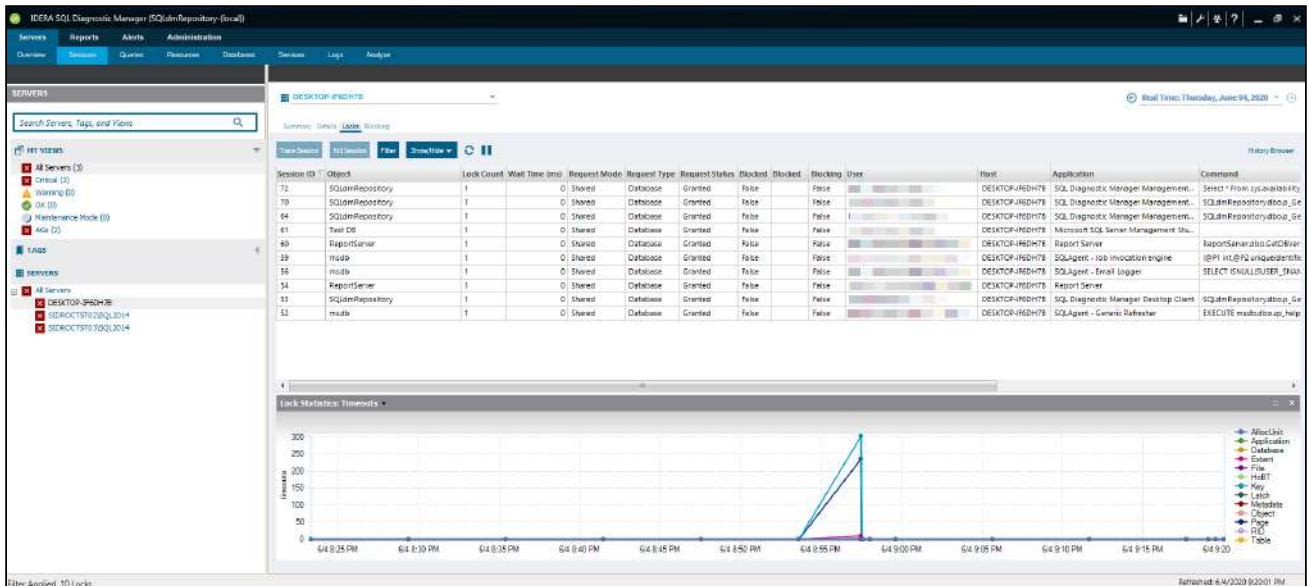
When you group data by multiple columns, be sure to select the headings in the hierarchy in which you want SQL Diagnostic Manager to display the results. After selecting the **Group By Box** check box, click, hold, and drag the first column heading (first order of grouping) into the grouping area. Next, click, hold, and drag the second column heading (second order of grouping) into the grouping area. Continue this process until you have the data hierarchy you want. Click the column headings to toggle sort order between ascending and descending.



You can remove a column from the grouping area by performing a click, hold, and drag of the column back into the grid area or by right-clicking the heading, and selecting **Remove This Column**.

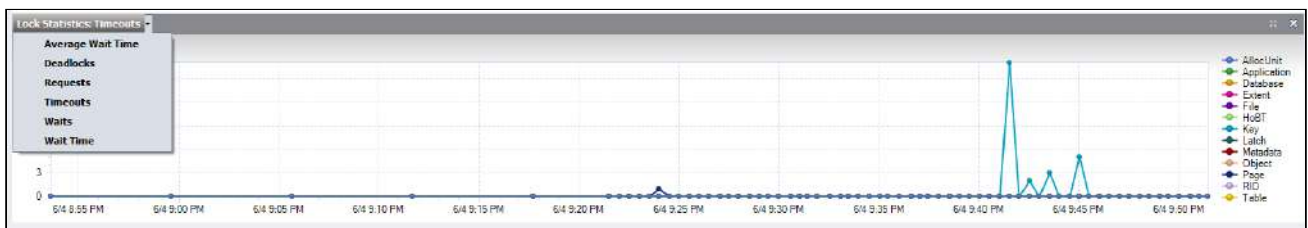
Analyze locked sessions

On the Locked Sessions view, you can see which problem locks are causing data availability problems and unacceptable responses. By combining the information provided on the Problem Locks view (including the type of Lock) with the **Lock Statistics: Waits** chart on the Sessions Summary tab, you can pinpoint, correct, or reschedule automated applications that cause extensive row, table, or database locking.



Use the drop-down list in the chart title to select charts that include:

- Average Wait Time
- Deadlocks
- Requests
- Timeouts
- Waits
- Wait Time information



Right-click any of these charts and either print, save as image, or export them to Microsoft Excel. In addition, you can select **Toolbar** for advanced customization options such as changing the chart color scheme and the type of chart shown.

Access the Locks view

You can open the Locks view of the SQL Diagnostic Manager Sessions tab by selecting the appropriate SQL Server instance, and clicking **Sessions > Locks**.

Available SQL Server lock objects

The Lock Statistics chart includes the following objects:

AllocUnit

Represents a lock on an allocation unit.

Application

Represents a lock on an application-specific resource.

Database

Represents a lock on a database, including all objects in the database.

Extent

Represents a lock on a contiguous group of eight pages.

File

Represents a lock on a database file.

HoBT

(Heap or BTree) Represents a lock on a heap of data pages, or on the BTree structure of an index.

Key

Represents a lock on a row in an index.

Latch

Represents a lock on a latch.

Metadata

Represents a lock on a piece of catalog information, also called metadata.

Object

Represents a lock on an item such as a table, stored procedure, or view, including all data and indexes. The object represents anything that has an entry in `sys.all_objects`.

Page

Represents a lock on an 8 KB page in a database.

RID

(Row ID) Represents a lock on a single row in a heap.

Table

Represents a lock on a table.

Analyze blocked sessions

The Blocking view provides a blocking tree, along with the lead blockers, for the selected SQL Server instance. A lead blocker is a session that blocks at least one session, which can then block other sessions.

- ✔ You can use the History Browser to [view past performance](#) and go back to the point in time to view the sessions involved with the blocking alerts you receive.

Navigate through the blocking tree to find sessions that cause blocks, and then either trace sessions to find out what is causing the blocks or stop the session to have it removed. To trace a blocked session or lead blocker select it from the list and click **Trace Session**. To stop a blocked session, or lead blocker, select it from the list, and click **Kill Session**.

- ✔ Deadlocks have no lead blockers and are circular in nature. For this reason, SQL Diagnostic Manager does not display deadlocks in the Blocking Tree but does display instances of deadlocks in the Blocking Chart. In addition, SQL Diagnostic Manager does not count self-blocking sessions as blocked or blocking sessions.

Access the Blocking view

You can open the Blocking view of the SQL Diagnostic Manager Sessions tab by selecting the appropriate SQL Server instance, and then clicking **Sessions > Blocking**.

Trace Session

When you click **Trace Session**, the Session Trace window opens. The Session Trace provides session statistics and information including the following:

- Overview of the session, including the CPU Time, Row Count, Lock Wait Timeout, Physical Reads and Write metrics.
- Last Command executed.
- All the SQL Server statements.

Blocking chart

The [Blocking chart](#) displays the number of blocked sessions, lead blockers, and total deadlocks at a given point in time. The total deadlocks represent the total number of deadlocks that have occurred since the last time the pane was refreshed.

You can right-click the chart and either print, save as image, or export this data to Microsoft Excel. In addition, you can select **Toolbar** for advanced customization options, such as changing the chart color scheme and the type of chart shown.

Blocking Sessions Report

The Blocking Sessions Report (Block Reports) displays blocking and blocked sessions' information. To access the Blocking Sessions Report in the Alert view, select the Show Block Details option in the right-click context menu. You can also access the same information on the **Blocking View** by selecting a specific SQL Server instance, pointing to **Sessions**, and then selecting the **Blocking** ribbon.

Key information provided for blocking and blocked sessions includes:

Blocking Process Details:

- Session ID
- Host Name
- User Name
- Application
- Database
- Last Batch Started
- Transaction ID
- Open Transactions

Blocked Process Details:

- Session ID
- Host Name
- User Name
- Application
- Database
- Last Batch Started
- Transaction ID
- Open Transactions
- Wait Time (ms)
- Wait Resource

This feature allows you to export statistics in XML format.

- ✔ To retrieve the blocking sessions report for SQL Diagnostic Manager, select the "**Non-Query activities>Capture Blocking (SQL 2005+)**" check box in the **Activity Monitor** Tab.

Deadlock Sessions Report

The Deadlock Sessions Report (Deadlock Reports) displays deadlock sessions information. To access the Deadlock Sessions Report in the Alert view, select the **Show Block Details** option in the right-click context menu. You can also access the same information on the **Blocking View** by selecting a specific SQL Server instance, pointing to **Sessions**, and then selecting the **Blocking** ribbon.

Key information provided for deadlock sessions includes:

Deadlock Process Details:

- Session ID
- Status
- Host Name
- User Name
- Execution Context
- Application
- Database
- Last Batch Started
- Last Batch Completed
- Last Transaction Started
- Transaction ID
- Transaction Name
- Open Transactions
- Wait Time (ms)

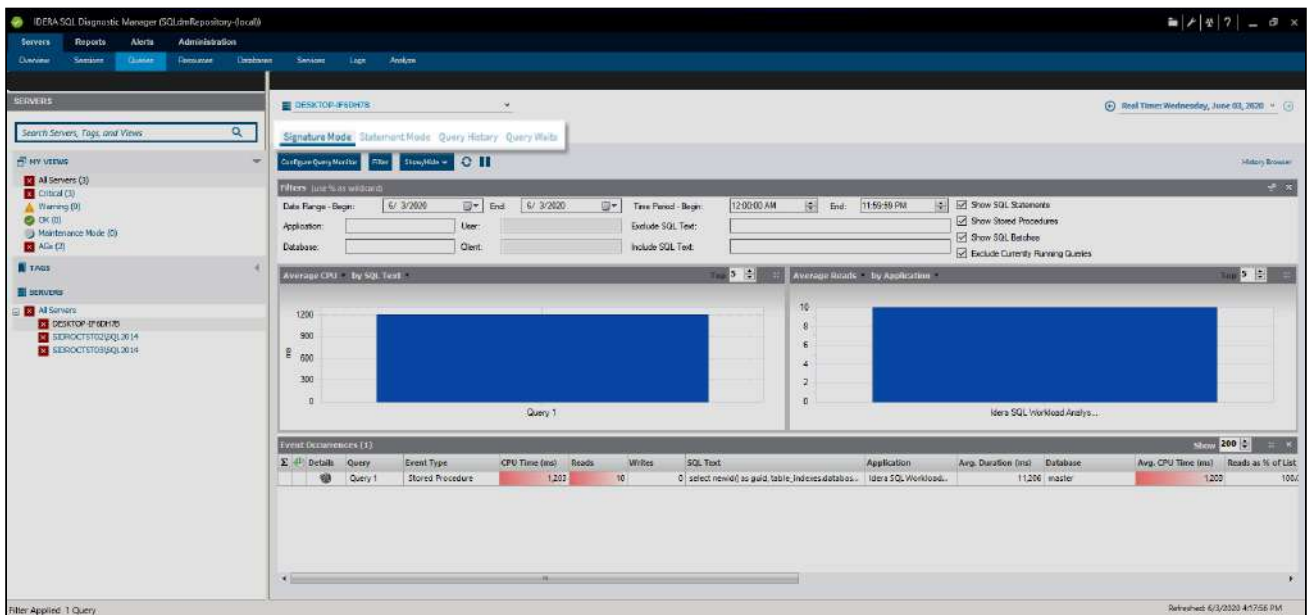
- Wait Type
- Wait Resource

This feature allows you to export statistics in XDL format.

- ✔ To retrieve the deadlock sessions report for SQL Diagnostic Manager, select the "**Non-Query activities>Capture deadlocks (SQL 2005+)**" check box in the **Activity Monitor** Tab.

Monitor query performance

Query Monitor is a useful SQL Server tool that allows you to log and analyze queries. Query Monitor allows you to establish criteria for the queries you wish to capture, and filter those results to analyze the queries returned. Use the Query Monitor when diagnosing poor query performance on your SQL Server instance. You should monitor queries if your SQL Server instance performance is poor, as a way to diagnose potential query issues.



- ✔ To explore query data in multiple ways and get a handle on queries in your system, try our [New advanced query views](#) of the SQL Diagnostic Manager web console.

Access the Queries tab

Access the Queries tab by clicking the appropriate monitored SQL Server instance, and then clicking the Queries tab.

Use the Queries tab

The Queries tab lists each event that occurs in the Query Monitor. For each event, among other details, SQL Diagnostic Manager lists the SQL text associated, start and end times, application, user information, and client computer associated with the event. Fields with SQL text hold 255 characters of text. To view the full SQL text, right-click the field and then select the appropriate option.

SQL Diagnostic Manager automatically assigns a name to each of your queries starting with Query 1. The name provides context to your query when switching from [Signature Mode](#), [Statement Mode](#), [Query History](#), and [Query Waits](#) views.

- ℹ Query names are not persistent. The names are different on different desktops and in different sessions on the same desktop.

The **Event Occurrences** chart at the bottom of the view displays the number of T-SQL statements, stored procedures, and SQL batches captured in the trace.

- ✓ Right-click any column heading to pick additional columns to display.

The Sigma (Σ) character indicates that the data in this row is at least partially composed of aggregated data. This means that the data is condensed into a day-length summary and some precision about the individual statements, client computers, and users is suppressed. Rows in this view with the Sigma character may include a mix of aggregated and non-aggregated data.

Understand the difference between Signature Mode and Statement Mode

Signature and Statement modes are two different ways to group and view your queries. You can toggle between each mode to see the differences.

Signature Mode (default Queries view)

All active queries are automatically reduced to the signatures (parameters are stripped) and grouped accordingly.

- ✓ You can also view signatures in the [Advanced query signature view](#) of the SQL Diagnostic Manager web console.

Statement Mode

All queries are presented exactly as they are collected by the Query Monitor Trace.

Use Filter Settings to customize your view

Use the Filters section of the Signature Mode, Statement Mode, and Query History views to pick the time range, what to include in the table, and even specific applications, client computers, databases, SQL text, and users to view.

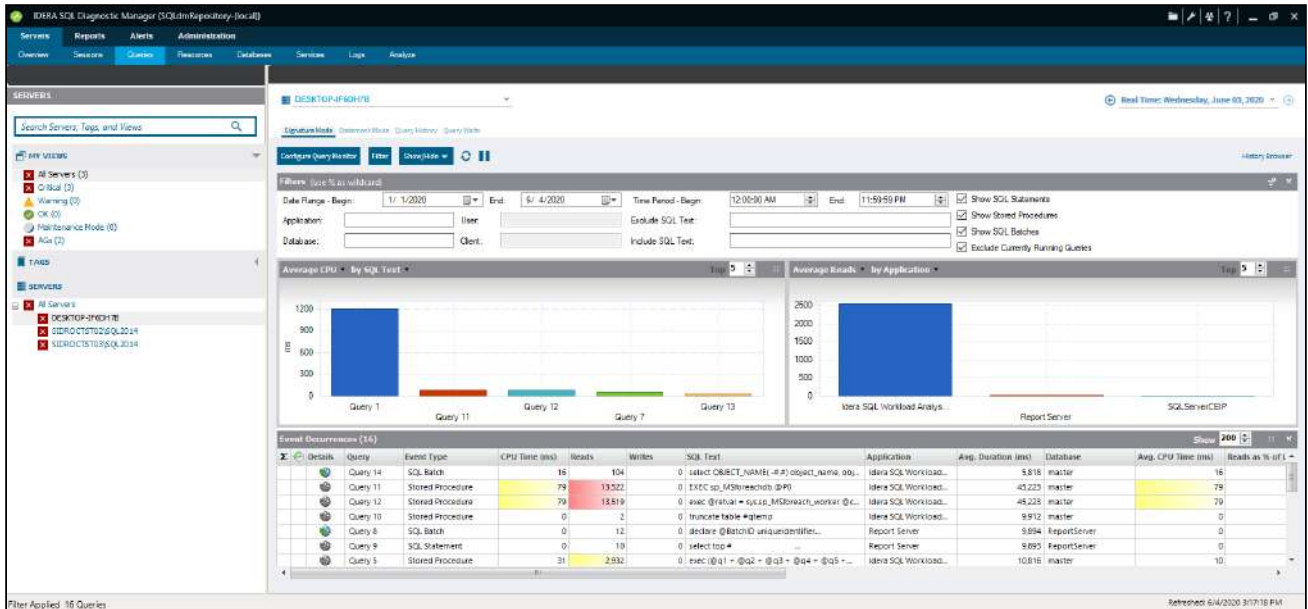
You can access advanced filters by clicking the **Filter** button. These options include the [ability to filter out specific applications, databases and SQL text from your query collection](#).

- ✓ The **Include incomplete queries** filter in the Signature Mode and Statement Mode views is disabled by default. If you would like to include data with null values in these views, select the **Include incomplete queries** check box in the Filters section.

- ✓ You can hide and unhide the Filter and List areas of the Query Monitor views in the ribbon.

View the query monitor signature mode

The Query Monitor Signature Mode view includes all active query information, automatically reduced to the signatures and grouped accordingly. Note that the parameters are stripped.





- ✓ If you want to prevent certain queries from appearing on the Query Monitor views, click [Configure Query Monitor > Advanced](#) and enter the applications, databases, or SQL text you want to exclude from the Query Monitor collection. You can also configure these options directly in the [SQL Diagnostic Manager web console](#).

The Query Monitor Signature Mode view is the place to start when investigating query performance. It broadly defines queries and trends with a less overwhelming amount of data and allows you to diagnose a query in a general sense. Once you identify a potential problem, you may want to view the [statement mode](#) view to see why a query is experiencing performance problems for a particular user, and then to the [query history](#) to see how the query performs over time and the associated trends.


The Signature Mode view is displayed as follows:

- **Filter** - Use Query filters to narrow the results to only those that most interest you. Do this by selecting the Date Range - Begin and End, the Application, User, Database, Client, and the SQL Text you want to include or exclude from your results.
- **Charts** - The primary focus of the Signature Mode view is in the charts. The chart type drop-down allows you to pick a chart type: Average CPU, Average Reads, Average Writes, Average Waits, Deadlocks, Blocking, CPU Per Second, and I/O Per Second. Also, you can choose to view the type of chart by the following categories: By SQL Text, By Application, or By Database.
- **Event Occurrences** - It lists each occurrence of the query and the associated statistics, such as the Event Type, CPU Time (ms), Reads, Writes, SQL Text, Application, Avg. Duration (ms), Database, Avg. CPU Time, Reads as % of List, Avg. Writes, CPU as % of List, Writes as % of List, Avg. I/O Per Sec, Avg. Wait Time (ms), Avg. CPU Per Sec, Avg. Reads, Occurrences, Most Recent Completion, Avg. Blocking Time (ms), and Avg. Deadlocks.

In the Event Occurrences, each listed query has the option to view the [Query Details](#) , click  to open the Query Details window (if the icon  is not colored, it means that there is no plan collected).

 **Sigma (Σ) character**

The Sigma character indicates that the data in this row is at least partially composed of aggregated data. This means that the data is condensed into a day-length summary and some precision about the individual statements, client computers, and users is suppressed. Rows in this view with the Sigma character may include a mix of aggregated and non-aggregated data.

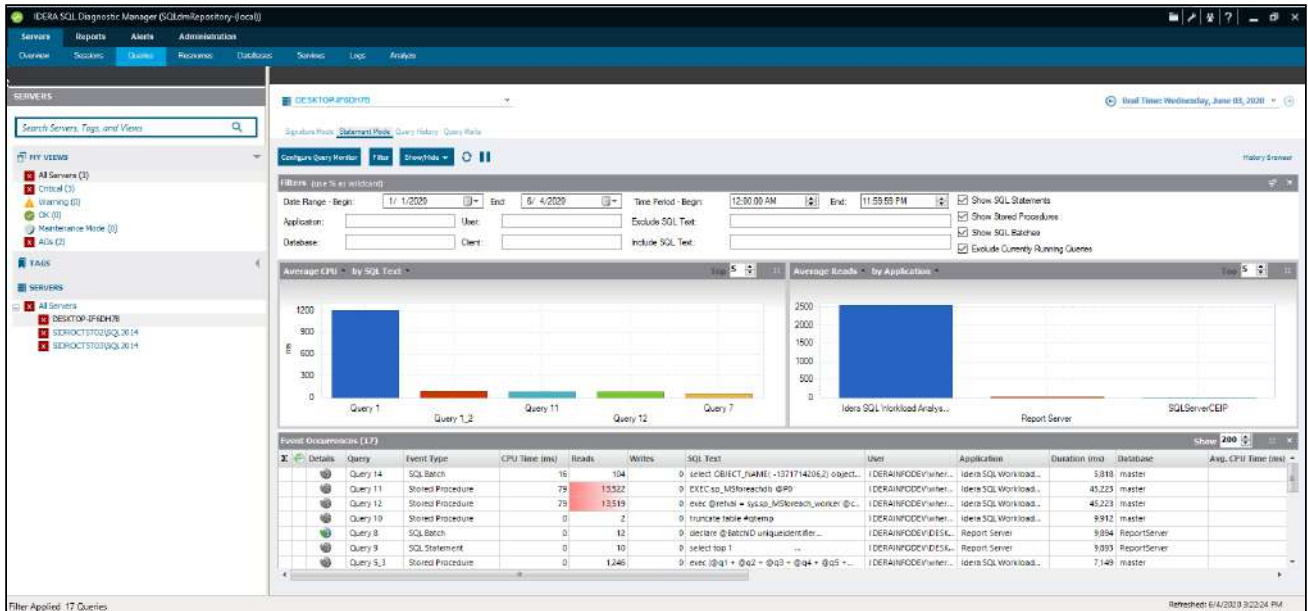
 In the Signature Mode view, right-click any of the listed queries under the Event Occurrences list and click Show Query Text to open the SQL Text window opens (In this window run a [Query Diagnosis](#) and receive a list of performance recommendations).

 **Understanding cell colors**

Red in the column indicates that the data in that row represents 20% or more of the total data displayed in the list, while yellow indicates that the data in the row represents 5% or more of the total data displayed in the list.

View the query monitor statement mode

The Statement Mode view includes queries that are presented exactly as they are collected by the Query Monitor Trace.






- ✓ If queries appear on the Query Monitor views that you never want to appear, click [Configure Query Monitor > Advanced](#) and enter the applications, databases or SQL text you want to exclude from the Query Monitor collection.

The Statement Mode view is where you go to drill-down into a specific execution of a particular query. Whereas the [signature mode view](#) provides a filtered look at your queries, the Statement Mode view provides all the detail you may need to diagnose a specific problem with a query. Once you find the problem query, History mode allows you to track the query over time and view trends.


The Statement Mode view is displayed as follows:

- **Filter** - Use Query filters to narrow the results to only those that most interest you. Do this by selecting the Date Range - Being and End, Application, User, Database, Client, and the SQL Text you want to include or exclude from your results.
- **Charts** - The primary focus of the Statement Mode view is in the charts. The charts type drop-down allows you to pick a chart type to display: Average Duration, Average CPU, Average Reads, Average Writes, Average Waits, Deadlocks, Blocking, CPU Per Second, and I/O Per Second. Also, you can choose to view the type of charts by the following categories: By SQL Text, By Application, By Database, By User, and By Client.
- **Event Occurrences** - It lists each occurrence of the query and associated statistics, such as the Event Type, CPU Time (ms), Reads, Writes, SQL Text, User, Application, Duration (ms), Database, Avg. CPU Time (ms), Avg. Writes, Avg. I/O per sec, Wait Time (ms), Avg. CPU Per Sec, Avg. Reads, Start Time, End Time, Blocking Time (ms), Deadlocks, Client, and Speed.

In the Event Occurrences, each listed query has the option to view the [Query Details](#), click  to open the Query Details window (if the icon  is not colored, it means that there is no plan collected).

 Sigma (Σ) character

The Sigma character indicates that the data in this row is at least partially composed of aggregated data. This means that the data is condensed into a day-length summary and some precision about the individual statements, client computers, and users is suppressed. Rows in this view with the Sigma character may include a mix of aggregated and non-aggregated data.

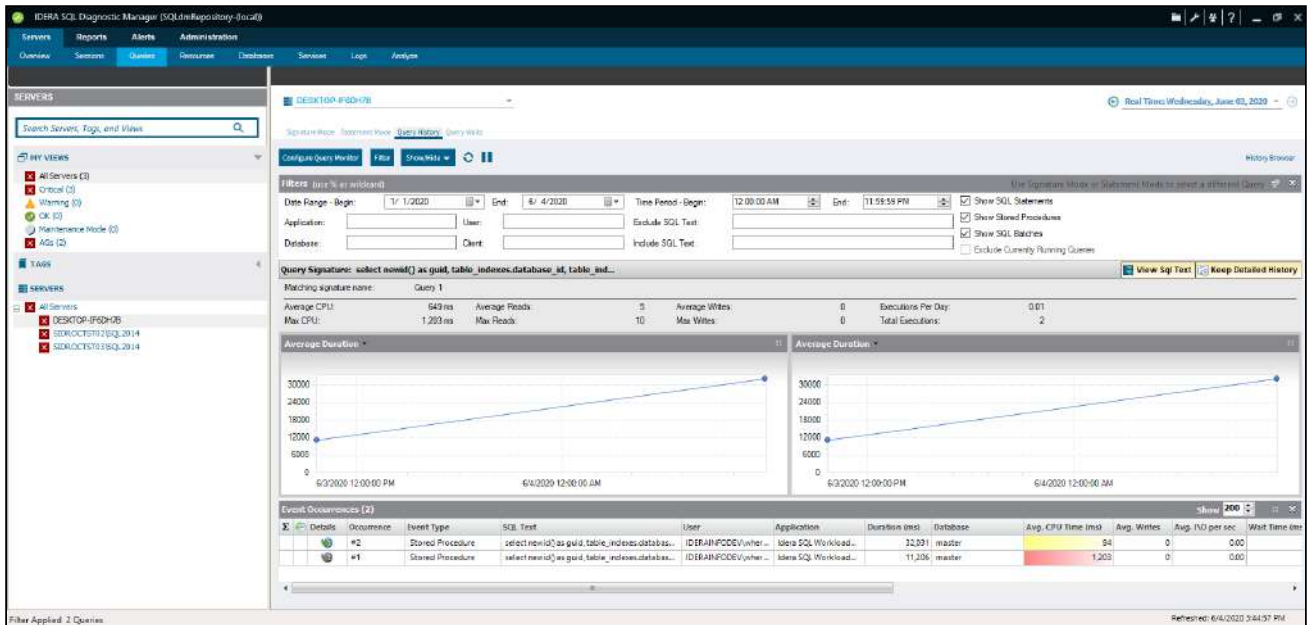
 In the Signature Mode view, right-click any of the listed queries under the Event Occurrences list and click Show Query Text to open the SQL Text window opens (In this window run a [Query Diagnosis](#) and receive a list of performance recommendations).

 Understanding cell colors

Red in the column indicates that the data in that row represents 20% or more of the total data displayed in the list, while yellow indicates that the data in the row represents 5% or more of the total data displayed in the list.

View the query history

Query History allows you to track the performance of a query over time. To access the Query History view, select the query you want to see on the query [statement mode](#) or [signature mode](#) view in the graph. You can also access the Query History view by right-clicking a session on the Session Details view or a Statement on the Query Waits view and selecting Show Query History.



Access the Query History view

You can open the Query History view of the SQL Diagnostic Manager Queries tab by selecting the appropriate SQL Server instance, and clicking **Queries > Query History**.


View Query History

1. Use the drop-down menu on each chart to view the history of each of the metrics associated with the query: Average Duration, Average CPU, Average Reads, Average Writes, Average Waits, Deadlocks, Blocking, CPU Per Second, and I/O Second.
2. Click **View SQL Text** to see the associated query text.
3. Click **Aggregate History** to keep an aggregated history for the selected query.

Use Query filters to narrow the results to only those that most interest you. Do this by selecting the data and time range, the application, user, database, workstation and even the SQL text you want to include or exclude from your results.

Event Occurrences is a list of each occurrence of the query and the associated statistics, such as the duration, CPU time, reads, writes, and associated SQL text. Click the Maximize window icon to view a larger version of this list.

- ✓ If Query Monitor is disabled but Activity Monitor is enabled, SQL statements appearing in the Event Occurrences grid come from the Blocking Session collector. The collector only populates the following fields: Occurrence, SQL text, Event Type, and Deadlocks.

 **Sigma (Σ) character**

The Sigma character indicates that the data in this row is at least partially composed of aggregated data. This means that the data is condensed into a day-length summary and some precision about the individual statements, client computers, and users is suppressed. Rows in this view with the Sigma character may include a mix of aggregated and non-aggregated data.

 **Understanding cell colors**

Red in the column indicates that the data in that row represents 20% or more of the total data displayed in the list, while yellow indicated that the data in the row represents 5% or more of the total data displayed in the list.

View query waits

The Query Waits view displays the queries on your SQL Server instance with the longest wait times. By default, the chart at the bottom of the view displays the query waits over time and allows you to further investigate based on Statements, Applications, Databases, Clients, Sessions, and Users.

By analyzing these waits you can better determine where your biggest bottlenecks are occurring and what changes could potentially have the biggest performance boost on your SQL Server instance.

You can select to view query waits as the waits appear over time or by duration of the wait, color coded by query. The chart includes each of the following wait type categories: [Backup](#), [I/O](#), [Lock](#), [Memory](#), [Non-I/O Page Latch](#), [Non-Page Latch](#), [Transaction Log](#), and [Other](#).


Access the Query Waits view

To open the Query Waits view of the SQL Diagnostic Manager Queries tab, select the appropriate SQL Server instance, select **Queries**, and click **Query Waits**.

Query Details

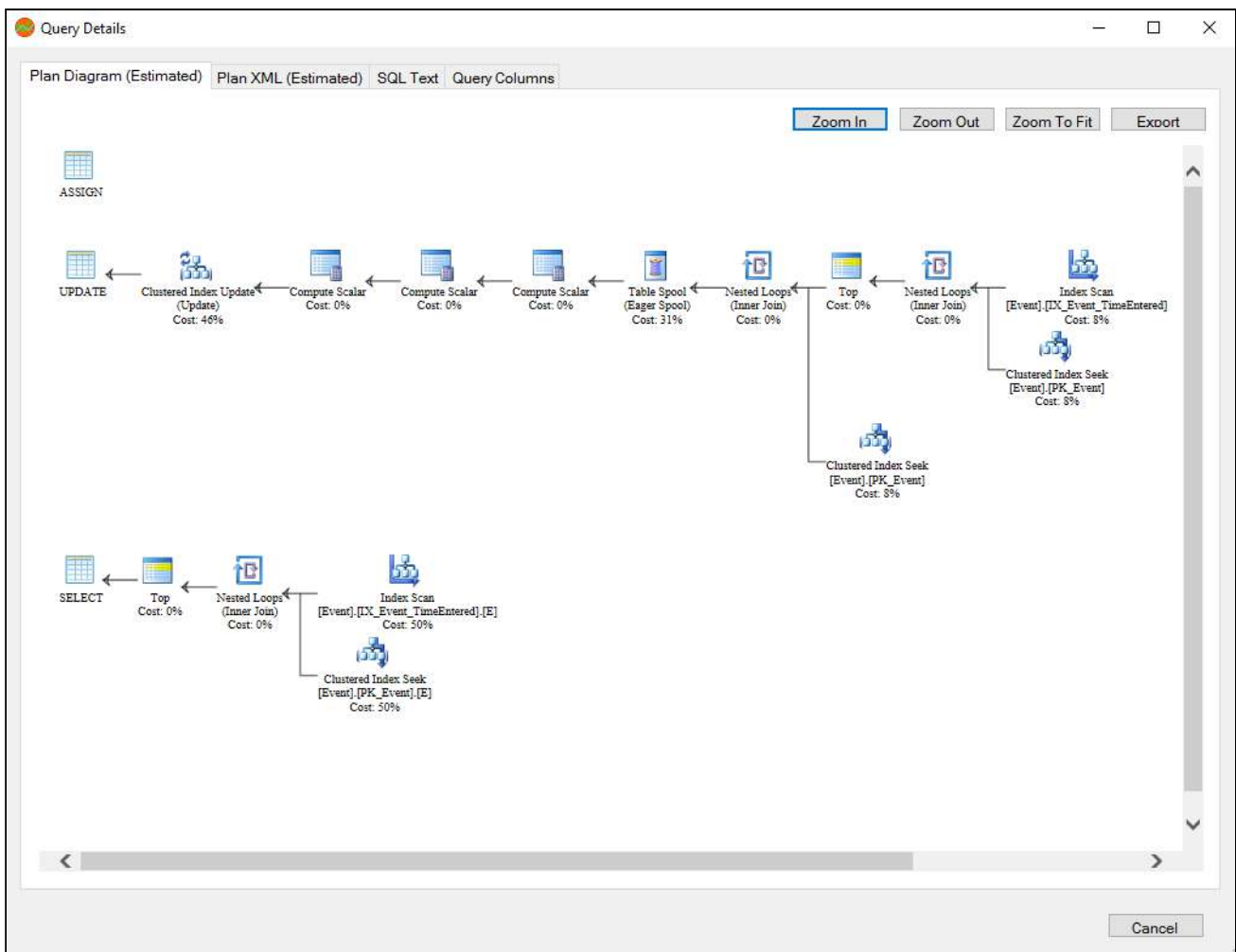
The Query Details window provides various views with detailed information on each query. The Query Details view is comprised of the following tabs:

- [Plan Diagram](#)
- [Plan XML](#)
- [SQL Text](#)
- [Query Columns](#)

 The Plan Diagram and Plan XML views are only available for instances running SQL Server 2008+.


Plan Diagram

The Plan Diagram (estimated) is a visual representation of the query execution plan (actual or estimated) available in XML format in SQL Server. The Plan Diagram displays a view of the tree of operations that make up a query. This tree shows individual operation nodes, pertaining graphical execution plan icon, along with basic information such as operator name and operation percentage of total cost.



In the Plan Diagram view you can perform the following actions:

- Select individual operators and view the set of [properties for each operator](#).
- Increase the size of the image (**Zoom in**).
- Reduce the image size (**Zoom out**).
- Select a size for the view so that fits the current view size (**Zoom to fit**).
- Export image of Plan Diagram.

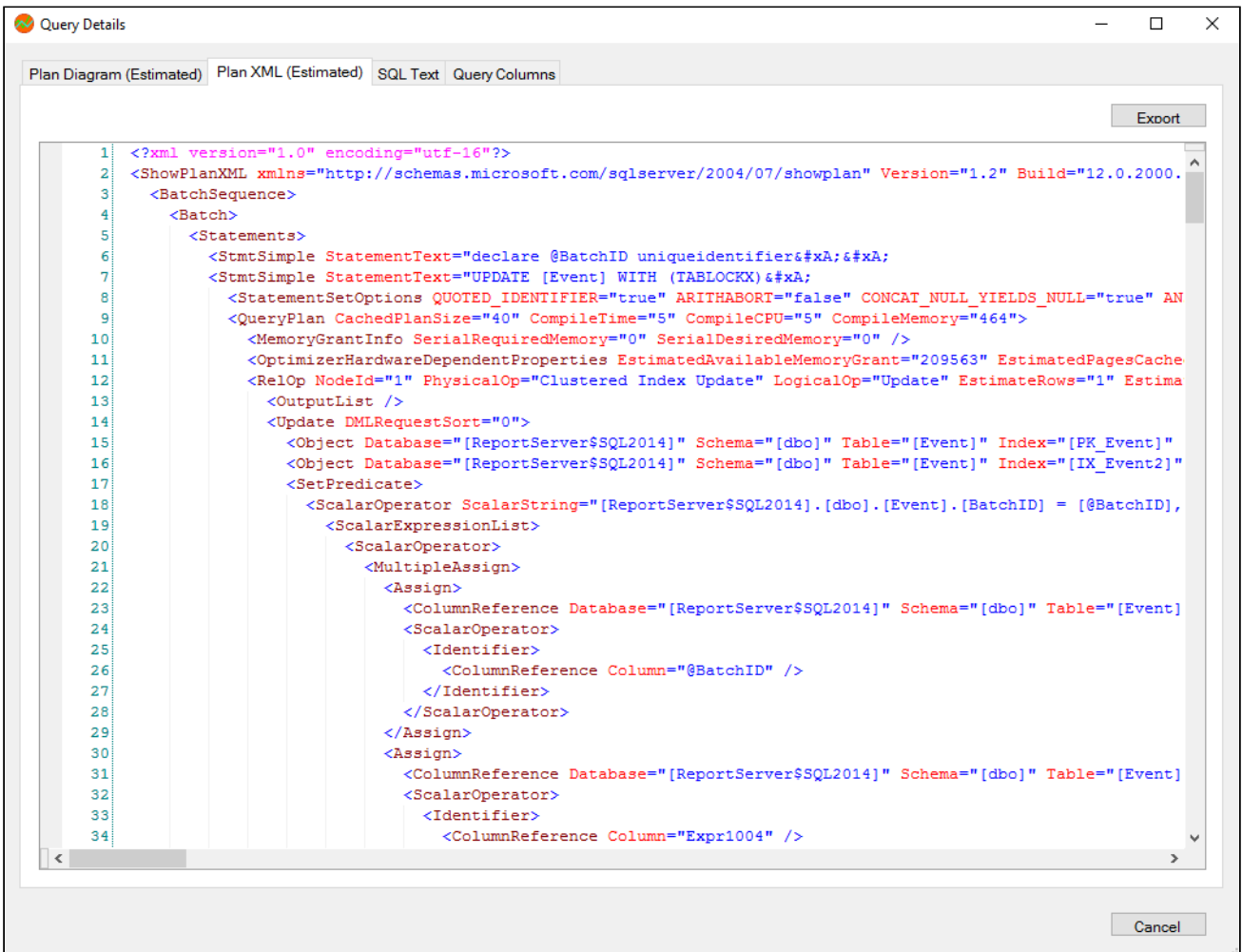
 The top three operator based on the percentage of total cost in a Plan Diagram are highlighted in yellow.

Estimated Query Plans

SQL Diagnostic Manager displays information on whether the plan displayed is an estimated query plan or an actual one.

Plan XML

The Plan XML of the Query Details window displays the actual XML of the query execution plan. It has a syntax-highlighting (color-coded) XML viewer.



```

1 <?xml version="1.0" encoding="utf-16"?>
2 <ShowPlanXML xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showplan" Version="1.2" Build="12.0.2000.
3 <BatchSequence>
4 <Batch>
5 <Statements>
6 <StmtSimple StatementText="declare @BatchID uniqueidentifier&#xA;&#xA;
7 <StmtSimple StatementText="UPDATE [Event] WITH (TABLOCKX)&#xA;
8 <StatementSetOptions QUOTED_IDENTIFIER="true" ARITHABORT="false" CONCAT_NULL_YIELDS_NULL="true" AN
9 <QueryPlan CachedPlanSize="40" CompileTime="5" CompileCPU="5" CompileMemory="464">
10 <MemoryGrantInfo SerialRequiredMemory="0" SerialDesiredMemory="0" />
11 <OptimizerHardwareDependentProperties EstimatedAvailableMemoryGrant="209563" EstimatedPagesCache
12 <RelOp NodeId="1" PhysicalOp="Clustered Index Update" LogicalOp="Update" EstimateRows="1" Estima
13 <OutputList />
14 <Update DMLRequestSort="0">
15 <Object Database="[ReportServer$SQL2014]" Schema="[dbo]" Table="[Event]" Index="[PK_Event]"
16 <Object Database="[ReportServer$SQL2014]" Schema="[dbo]" Table="[Event]" Index="[IX_Event2]"
17 <SetPredicate>
18 <ScalarOperator ScalarString="[ReportServer$SQL2014].[dbo].[Event].[BatchID] = [@BatchID],
19 <ScalarExpressionList>
20 <ScalarOperator>
21 <MultipleAssign>
22 <Assign>
23 <ColumnReference Database="[ReportServer$SQL2014]" Schema="[dbo]" Table="[Event]"
24 <ScalarOperator>
25 <Identifier>
26 <ColumnReference Column="@BatchID" />
27 </Identifier>
28 </ScalarOperator>
29 </Assign>
30 <Assign>
31 <ColumnReference Database="[ReportServer$SQL2014]" Schema="[dbo]" Table="[Event]"
32 <ScalarOperator>
33 <Identifier>
34 <ColumnReference Column="Expr1004" />

```

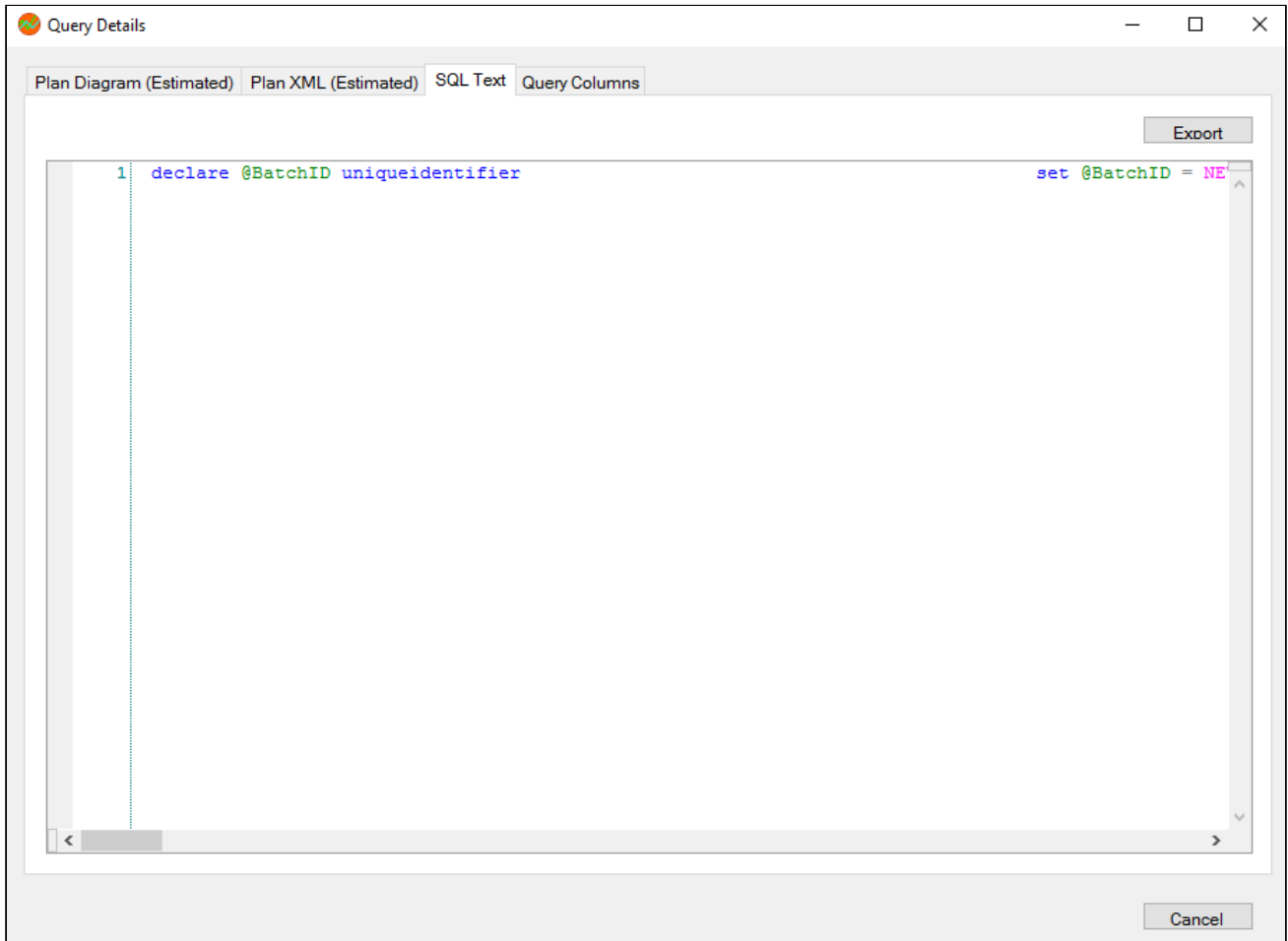
In the Plan XML view you can perform the following actions:

- Export and save XML file for review in other application such as SSMS or email query execution plan to other DBAs.

Save the file with **.sqlplan** extension to view the graphical query plan in SSMS.

SQL Text

The SQL Text tab of the Query Details window shows the underlying SQL Text for the query execution plan.

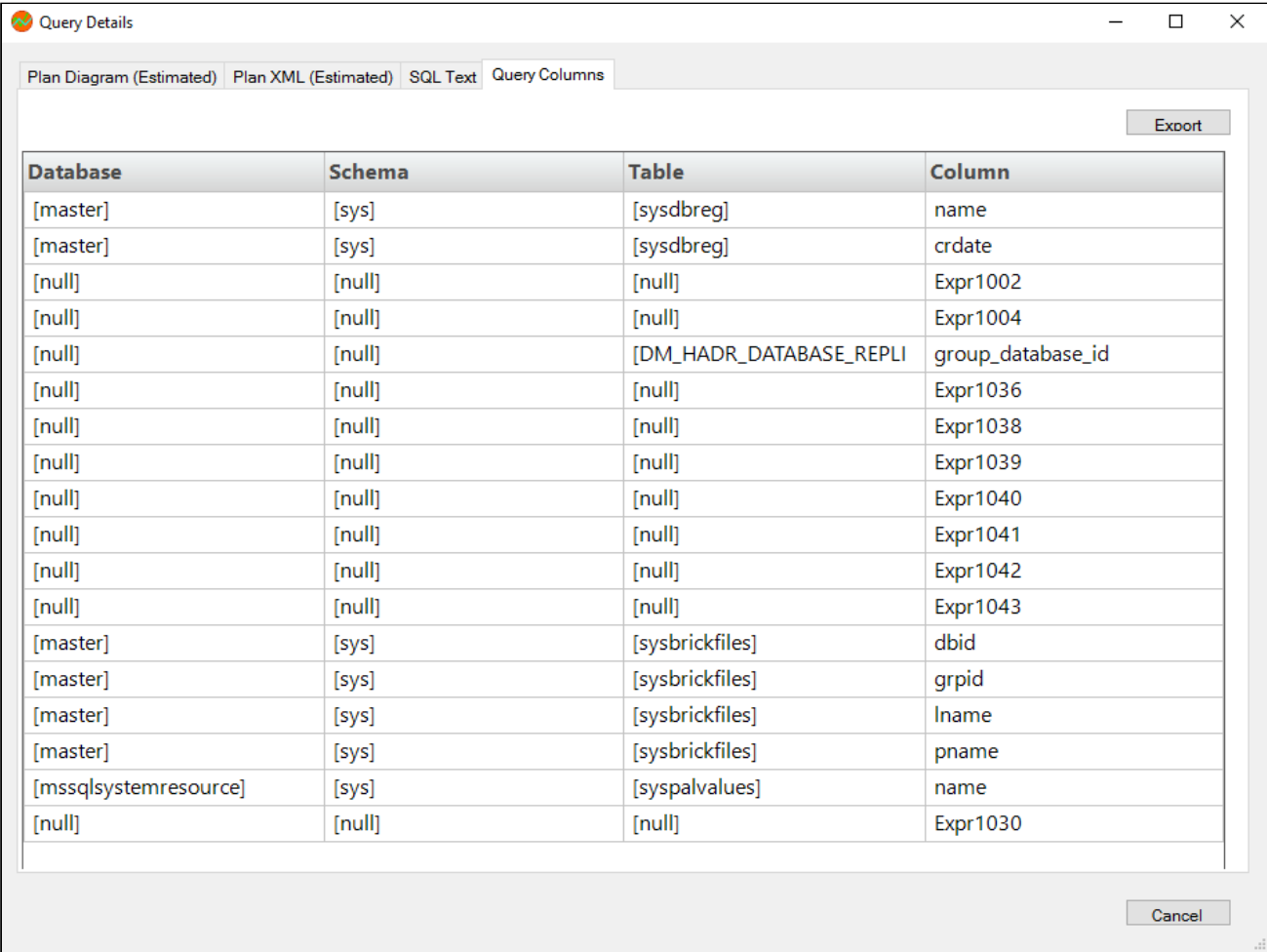


In the SQL Text view you can perform the following actions:

- Export and save SQL Text for review.
- Access the Plan Diagram, Plan XML, and Query Columns tabs.

Query Columns

The Query Columns tab of the Query Details window shows all the referenced columns for the query execution plan.



In the Query Columns view you can perform the following actions:

- Export and Save information to a text file.

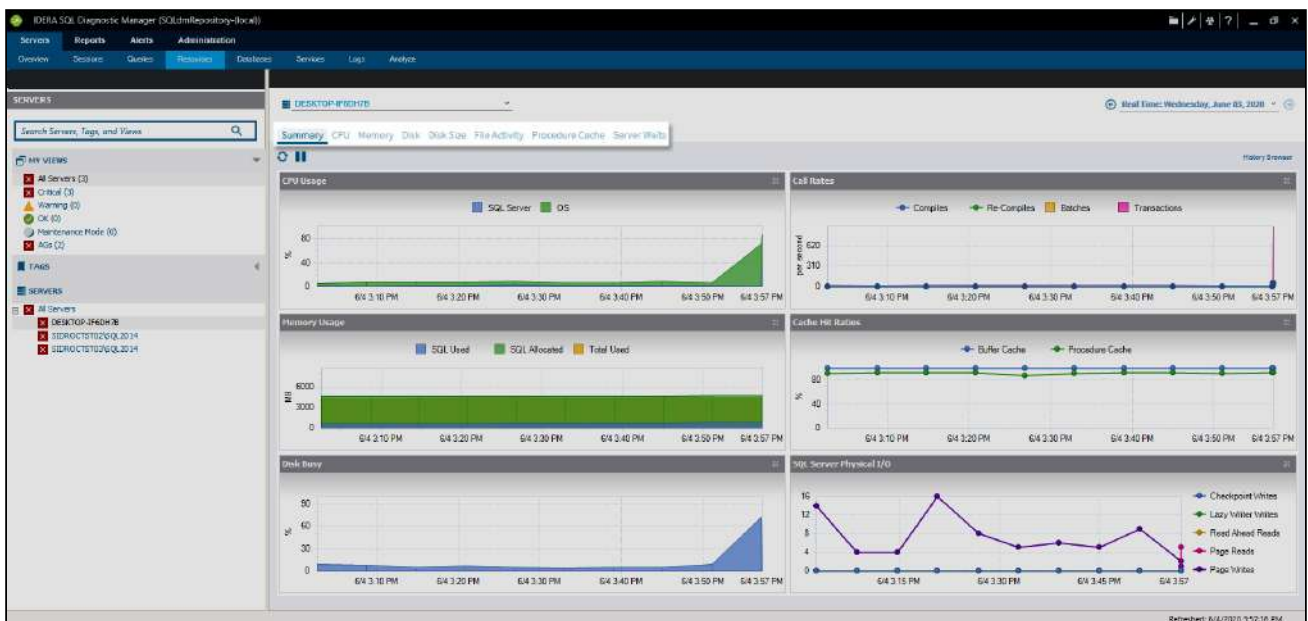
Monitor resources

The Resources tab allows you to monitor the system resources on the computer hosting the SQL Server instance as well as what used by SQL Server.

Resource bottlenecks are often the cause of SQL Server performance problems. These bottlenecks can result from poor database application design or point to other system processes using valuable resources needed by SQL Server.

Access the Resources tab

Access this tab by clicking the appropriate monitored SQL Diagnostic Manager and selecting the Resources tab.



Alerts associated with the Resources tab

The following alerts are associated with the Resources tab:

Disk Reads / Seconds Per Disk

The number of disk reads per second, per disk.

Disk Transfers / Seconds Per Disk

The number of disk transfers per second, per disk.

Disk Writes / Seconds Per Disk

The number of disk writes per second, per disk .

OS Average Disk Queue Length

The average number of both read and write requests queued for all the disks on the computer hosting the SQL Server instance.

OS Disk Time (Percent)

Indicates the percentage of elapsed time that all the disks were busy servicing read and write requests on the SQL Server computer.

OS Memory Usage (Percent)

Indicates the percentage of total memory used on the SQL Server computer.

OS Paging (Per Second)

Indicates the page fault rate for all processes on the SQL Server computer.

OS Privileged Time (Percent)

Indicates the percentage of CPU time used by all processes on the computer hosting the SQL Server instance, executed in Privileged mode.

OS Processor Queue Length

Is the number of ready threads in the processor queue on the computer hosting the SQL Server instance.

OS Processor Time (Percent)

Indicates the percentage of CPU time used by all processes on the computer hosting the SQL Server instance.

OS User Time (Percent)

Indicates the percentage of CPU time spent by all processes, on the computer hosting the SQL Server instance, executing in User mode.

Page Life Expectancy

The number of seconds a page stays in the buffer pool without references.

Procedure Cache Hit Ratio (Percent)

The ratio of procedure cache hits to procedure cache lookups. This indicates when an execution plan is reused from memory as opposed to compiled from disk.

SQL Server CPU Usage (Percent)

Indicates the percentage of the total CPU used by the SQL Server instance.

SQL Server Memory Usage (Percent)

Indicates the percentage of total memory used by SQL Server.

Get the resource performance summary

The Resources Summary tab displays real-time charts that allow you to quickly view key diagnostic statistics for monitoring the resources for the selected SQL Server instance and computer on which it resides. The Resources Summary tab is especially useful in determining resource bottlenecks. For more details, click any of the charts to navigate to the associated tab.

Access the Resources Summary view

You can open the Summary view of the SQL Diagnostic Manager Resources tab, select the appropriate SQL Server instance, select **Resources**, and click **Summary**.

Use the CPU Usage chart

The [CPU Usage chart](#) helps you determine the amount of strain your SQL Server instance is putting upon the CPU. This is an important factor in analyzing poorly-performing SQL Server instances. Click this chart to open the CPU, which contains more charts and information on your CPU resources.

- ✔ SQL Server is the primary process consuming CPU on the computer hosting your instance. If this is the case and the total CPU usage is consistently higher than 80 percent, the computer requires your attention. High usage may indicate the need to upgrade your CPU or add more processors.

Use the Call Rates chart

The [Call Rates chart](#) helps you determine why you have high CPU usage. The Call Rates chart displays the following metric statistics:

Batches

Displays the rate at which SQL Server issues SQL Statement batches.

Compiles

Displays the rate at which SQL Server compiles SQL Statements into the Procedure Cache.

Re-Compiles

Displays the rate at which SQL Server recompiles stored SQL Statements into the Procedure Cache.

- ✔ If the number of re-compiles is consistently higher than 10, or the number of compiles is high, you may need to tune your queries. Consider running the [Query Monitor](#) as a way of diagnosing query performance issues. You can enable the Query Monitor on the Query Monitor tab of the Monitored SQL Server Properties window.

Transactions

Displays the number of SQL Statement transactions affecting the Procedure Cache.

Use the Memory Usage chart

The [Memory Usage chart](#) allows you to analyze the way in which your SQL Server instance uses memory. The chart provides the amount of memory allocated to SQL Server, the memory currently used by SQL Server, and the total amount used on the host computer.

- ✔ If the SQL Used metric is consistently high compared to the amount of physical memory on the computer where the SQL Server instance resides, it may indicate that you need more physical memory.

Use the Cache Hit Ratios chart

The [Cache Hit Ratios chart](#) allows you to view how often a request is serviced by the buffer and procedure cache.

- ✔ An acceptable Buffer Cache and Procedure Cache percentage is 90 percent or higher. If this number is consistently lower, analyze your memory settings and add more memory if necessary.

Use the Disk Busy chart

Use the [Disk Busy chart](#) to analyze bottlenecks in your file system.

- ✔ If your Disk Busy is consistently over 90 percent you may need to use a faster disk drive, move files to an additional disk or server, or add disks to a RAID array to improve your SQL Server performance.

Use the SQL Server Physical I/O chart

The [SQL Server Physical I/O chart](#) allows you to view the number of pages read from, and written to, the physical disk.

The SQL Server Physical I/O chart displays the following reads and writes:

Checkpoint Writes

Save time during recovery by creating a point at which all dirty pages are guaranteed written to disk.

Lazy Writer Writes

Create more available buffers for the system by clearing all unnecessary buffers.

Page Reads

Are disk writes where the requesting session must wait for the disk write operation to finish.

Page Writes

Are disks writes where the requesting session must wait for certain operations to finish.

- ✔ If the number of reads and writes approaches the limits of the I/O capacity of the computer the SQL Server instance resides, consider reducing the values by tuning your application or database, increasing the I/O capacity of your hardware, or adding memory.

Read Ahead Reads

Are part of a process in which SQL Server predicts what pages a session requires next and writes them to memory in advance.

Understanding your chart data and refresh interval

SQL Diagnostic Manager charts provide data whether collected real-time or as part of a scheduled collection event. For additional information about how SQL Diagnostic Manager displays your chart data, see [Charts](#).

Get CPU performance details

The CPU tab displays key CPU statistics that are updated according to your refresh interval. These statistics can help you determine the cause to any performance bottlenecks that are occurring on your SQL Server instance.

Access the CPU view

To open the CPU view of the SQL Diagnostic Manager Resources tab, select the appropriate SQL Server instance, select Resources, and click **CPU**.

CPU tab charts

The following [charts](#) are available in the CPU tab:

CPU Usage

Displays the percentage of a CPU consumed by your SQL Server, virtual machine, and host server. On multi-processor computers, CPU time used by SQL Server is derived from the processors available for use by SQL Server.

Processor Queue Length

Displays the number of threads in queue for processor time on the computer that hosts the monitored SQL Server instance.

Processor Time

Displays the percentage of elapsed time that is used by processor executing code in privileged mode.

Call Rates

Displays the rate of SQL Server batch statements, SQL compiles into procedure cache, and SQL Server recompiles of statements stored in the procedure cache.

Understanding your chart data and refresh interval

SQL Diagnostic Manager charts provide data whether collected real-time or as part of a scheduled collection event. For additional information about how SQL Diagnostic Manager displays your chart data, see [Charts](#).

Get memory performance details

The Memory tab displays the dynamic memory currently in use by SQL Server as well as how that memory satisfies the requirements of SQL Server components. For example, you can see what percentage of the total allocated memory is currently used by the procedure cache. You can control the amount of memory SQL Server uses by either setting a fixed memory value or setting minimum and maximum memory usage values.

Static structures, like procedure headers, use the amount of memory derived from subtracting the sum of the graphed sections displayed on the Memory tab from the total allocated memory, also provided on the Memory tab.

Access the Memory view

To open the Memory view of the SQL Diagnostic Manager Resources tab, select the appropriate SQL Server instance, click **Resources**, and select **Memory**.

Memory tab charts

The following charts are available on the Memory tab:

SQL Server Memory Usage

Displays the SQL Memory Usage by what is used by SQL Server compared to what is allocated and the total used.

VM/Host Memory Usage

Displays the memory usage by your virtual machine and host as a percentage of total memory available.

Paging

Displays the memory pages per second.

SQL Server Memory Areas

Displays the memory usage of the Procedure Cache, Connections, Locks, Database, Optimizer, Sort, Hash, and Index.

VM Memory Areas

Displays the amount of memory allocated to and used by SQL Server to the total memory consumed on this virtual machine.

Host Memory Areas

Displays the amount of memory allocated to and used by SQL Server to the total memory consumed on this host server.

Cache Hit Ratios

Displays the hit ratios of the Buffer and Procedure cache.

Buffer Cache

Displays the buffer cache used by Free and Active pages.

Procedure Cache

Provides the amount of allocated memory used by dynamic and static stored procedures. Included in this value is the memory used by server side cursors and plans. From the total memory allocated to SQL Server, the pinned tables, open objects, connections, locks, and free buffer values are subtracted. The remainder is divided between procedure and data cache.

Page Life Expectancy (sec)

Provides the time in seconds that a page remains in the cache or buffer pool.

Understanding your chart data and refresh interval

SQL Diagnostic Manager charts provide data whether collected real-time or as part of a scheduled collection event. For additional information about how SQL Diagnostic Manager displays your chart data, see [Charts](#).

Get disk performance details

The Disk tab allows you to view key statistics on the way your disks are used on the computer hosting your SQL Server instance. This allows you to analyze your disk performance for any performance bottlenecks that may be occurring.

Disk information is displayed in four charts that display the drives the computer on which the SQL Server instance is located. You can pick which four charts you view by selecting from the drop-down list at the top of each chart. You can select from the following charts:

✔ You must have [OS metrics enabled](#) to show the information in the charts on the Disks tab.

Access the Disk view

To open the Disk view of the SQL Diagnostic Manager Resources tab, select the appropriate SQL Server instance, click **Resources**, and select **Disk**.

Disk tab details

Disk Busy Total

Displays the percentage of time during the sample interval that the disk was not idle for all the disks(combined) on the server that the SQL Server instance resides.

Disk Busy Per Disk

Displays the percentage of time during the sample interval that each disk was not idle.

Average Disk Queue Length Total

Displays the average length of the disk queue during the sample interval for all disks (combined) on the server that the SQL Server instance resides.

Average Disk Queue Length Per Disk

Displays the average length of the disk queue during the sample interval for all disks (individual) on the server that the SQL Server instance resides.

SQL Server Physical I/O

Displays the number of read and write requests sent to the physical disk the SQL Server instance resides.

SQL Server Physical I/O Errors

Displays the number of errors sent from the physical disk the SQL Server instance resides.

Disk Reads/Second Per Disk

Displays the number of reads per disk per second during the sample interval.

Disk Transfers/Second Per Disk

Displays the number of transfers per disk per second during the sample interval.

Disk Writes/Second Per Disk

Displays the number of writes per disk per second during the sample interval.

Average Disk ms/Read Per Disk

Displays the average length of time (in milliseconds) for a disk to read during the sample interval.

Average Disk ms/Transfer Per Disk

Displays the average length of time (in milliseconds) for a disk transfer during the sample interval.

Average Disk ms/Write Per Disk

Displays the average length of time (in milliseconds) for a disk write during the sample interval.

VM Disk Usage

Displays the amount of data read from and written to disk in KB per second for this virtual machine during the sample interval. For additional information about how SQL Diagnostic Manager works with virtual machines and collects metrics, see [How SQL Diagnostic Manager works with a virtual environment](#).

Host Disk Usage

Displays the amount of data read from and written to disk in KB per second for the host server during the sample interval. For additional information about how SQL Diagnostic Manager works with virtual machines and collects metrics, see [How SQL Diagnostic Manager works with a virtual environment](#).

Understanding your chart data and refresh interval

SQL Diagnostic Manager charts provide data whether collected real-time or as part of a scheduled collection event. For additional information about how SQL Diagnostic Manager displays your chart data, see [Charts](#).

Get disk size details

The Disk Size tab allows you to view key statistics on the way your disks are used on the computer hosting your SQL Server instance. This allows you to analyze your disk performance for any performance bottlenecks that may be occurring.

- ✔ You must have [OS metrics enabled](#) to show the information in the charts on the Disks tab.

Access the Disk Size view

You can open the Disk Size view by selecting the appropriate SQL Server instance, and then clicking **Disk Size** in the **Views** group of the **Resources** tab. Click image to view full size.

Disk Size tab details include:

- Drive Name
- Total Size (MB)
- Used (MB)
- Free Disk (MB)
- SQL Data Used (MB)
- SQL Data Free (MB)
- SQL Log (MB)
- Other Files (MB)

Current Disk Usage chart

The Disk size tab includes a chart that displays the current disk usage. The drop-down list allows you to choose to view the data in megabytes or percentages.

Understanding your chart data and refresh interval

SQL Diagnostic Manager charts provide data whether collected real-time or as part of a scheduled collection event. For additional information about how SQL Diagnostic Manager displays your chart data, see [Charts](#).

- ✔ You can print, export to Excel, or save the Current Disk Usage chart image.

View file activity

The SQL Diagnostic Manager File Activity allows you to view the relative activity of your database files, on a file-by-file basis. Each database file is listed in a column that represents the disk or mount point on which the files reside. At the top of each column the overall Reads, Writes, and Transfers per second are listed. Click an individual file to expand it and view a graph and the activity data specific to that file.

For each database file, the following information is provided:


Reads and Writes Per Second

The numbers listed are the average number of reads and write operations per second over the period between collections. Collection times are configured on the Server Properties window.

Activity lights

Read (green) and Write (blue) lights flash with activity.

Activity graph

A bar graph under the file name displays the recent trend of reads and writes per second. This graph expands when you double-click the file. The expanded Activity graph displays the activity trend for the amount of time specified on the Console Options window ( > **Console Options**).

Activity data

The arrows and the associated data represent the amount of data activity occurring on the file and whether the amount of activity is rising or falling relative to the previous data point.

- ✓ Right-click a disk column to Expand/Collapse all files for a database, Expand/Collapse all files on a disk, Navigate to Disk view, or Navigate to the Disk Files view.

Access the File Activity view

To open the File Activity view of the SQLDM Resources tab, select the appropriate SQL Server instance, click **Resources**, and select **File Activity**.

Database Files + Remaining Disk Files

Database Files + Remaining Disk Files chart depicts the activity of all the non-SQL database files and files not included in the File Activity window.

Use the filter to display vital database activity

The filters at the top of the File Activity window allow you to limit the amount of data to that data which is most important to you.

- ✓ For an accurate filter, use the percent (%) wildcard character in the **File Path Like** or **File Name Like** field if you do not include the entire name.

Filter Drive

Use the **Drive** drop-down to select the drives to display on the File Activity window. You can also enter a file path in the **File Path Like** field to select a subset of drives. This field accepts the percent (%) wildcard character.

Filter Database

Use the **Database** drop-down to select the databases to display on the File Activity window. You can also enter a file name in the **File Name Like** field to select a subset of databases. This field accepts the percent (%) wildcard character.

File Type

Check the boxes of the file types to display on the File Activity window.

Sort By

Select to sort by **Database Name**, **File Name**, **Reads Per Second**, or **Writes Per Second**.

Sort Direction

Further, sort your data by selecting **Ascending** or **Descending** order.

Manage your procedure cache

The Procedure Cache view allows you to view each item cached, the size, and the SQL text associated. Graphs below the cache listing provide you with a breakdown of the key Procedure Cache statistics that can help you analyze performance bottlenecks on your SQL Server instance.

The **Usage** column displays the number of times the associated cache object is used since inception. The **References** column displays the number of other cache objects referencing the associated cache object, with "1" as the base.

Access the Procedure Cache view

You can open the Procedure Cache view of the SQL Diagnostic Manager Resources tab by selecting the appropriate SQL Server instance, and then clicking **Resources > Procedure Cache**.

Procedure Cache tab charts

The following [charts](#) are available on the Procedure Cache tab:

Cache Size by Object Type

Displays the amount of the procedure cache used by various objects including:

- Adhoc Query
- Check Constraint
- Default Constraint
- Extended Procedure
- Prepared Statement
- Replication Procedure
- Rule
- Stored Procedure
- System Table
- Trigger
- User Table
- View

Object Types

Use the drop-down button to select either Hit Ratio, Object Count, or Use Rate.

- Hit Ratio offers a chart of the cache hit ratios of the individual object types in memory through the last 25 refreshes. For example, you can view the hit ratio of the extended procedure object.
- Object Count displays the number of objects using the procedure cache.
- Use Rate displays the rate at which each of the object types is used in the procedure cache.

View server waits

The Server Waits view allows you to see all of the waits affecting server performance. By correlating the areas of the SQL Server instance that are waiting with other statistics, it is possible to narrow down possible causes of the waits. All waits occurring on your system are displayed in a graph that shows the time the wait happened and the length of time it took for the wait to clear. You can filter the graph by the wait type to view how a particular wait is affecting your server. Below the graph, SQL Diagnostic Manager displays a table that includes the wait category, wait type, length of time the wait took to clear, total wait time for all the collected wait types, and a definition of the wait type.

Server waits are always collected. History mode for this view is available.

Access the Server Waits view

You can open the Server Waits view of the SQL Diagnostic Manager Resources tab by selecting the appropriate SQL Server instance, and clicking **Resources > Server Waits**.

Available wait types

You can filter the view by [Backup](#), [I/O](#), [Lock](#), [Memory](#), [Non-I/O Page Latch](#), [Non-Page Latch](#), [Transaction Log](#), and [Other](#).

Available wait categories

You can select from the following wait categories:

Total Waits

Total waits include all the types of waits collected.

Signal Waits

Signal waits occur when a resource is now available (after a resource wait) and the task is waiting to run again.

Resource Waits

Resource waits occur when access to a resource is denied because the resource is in use or not available.

Monitor databases

SQL Diagnostic Manager allows you to monitor your database backups, restores, table sizes, table row counts, configurable options, and replication status. SQL Diagnostic Manager provides a graphical analysis of the used, unused, and index disk space of the database. You can also view the size and structure of database tables. SQL Diagnostic Manager provides a list of data size, text or image size, index page size, and the number of table rows, making reorganization and index creation decisions easy.

The following concepts are important to understand when using the Databases tab:

Percentage of Database/Log Full

The **%Data Full** and **%Log Full** columns on the Database Summary and Database Files views refer to the current space used as a percentage of the potential maximum available space. The larger the number the closer you are to running out of space. These columns can have associated alert thresholds.

Potential Maximum Available Space

The **Data Potential Growth** and **Log Potential Growth** columns on the Database Summary and Database Files views refer to the currently allocated space plus the available expansion space.

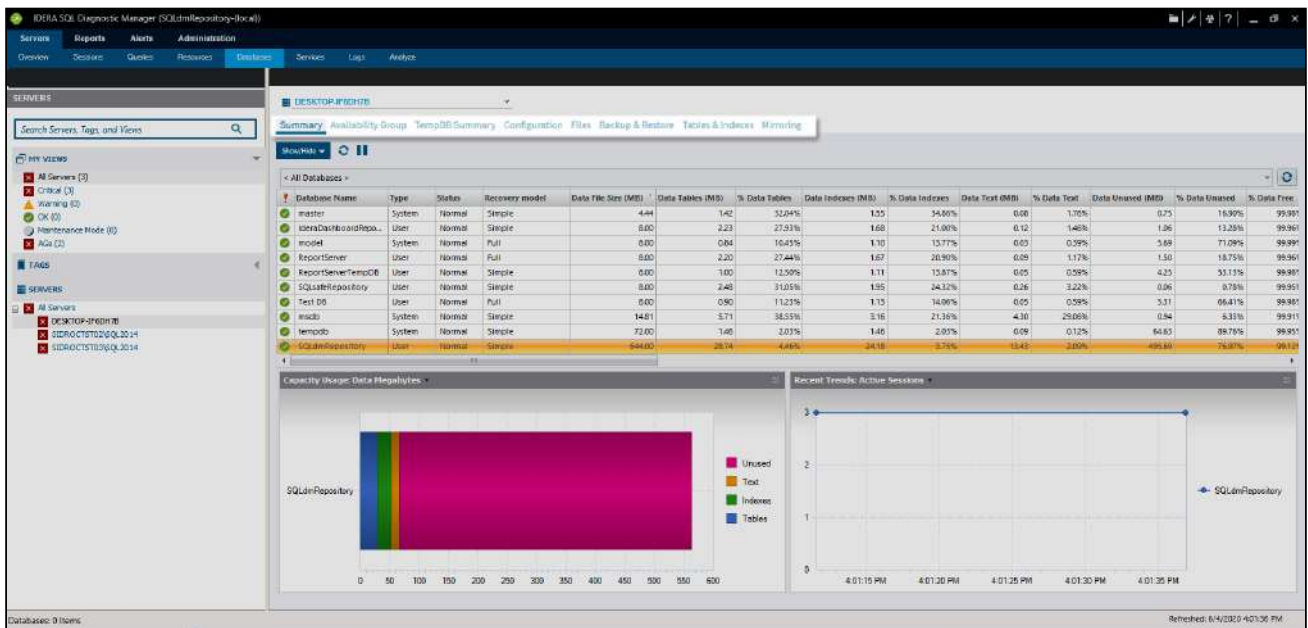
Available Expansion Space

Available Expansion Space refers to the available disk space that files can grow into as determined by the growth settings for the files, the location of the files, and the free space on those disks.

Multiple database files as well as files from other databases can potentially compete for space on the same disk. For example, if the data and log files for several databases are on the same disk and the data file of one database expands, the available expansion space for all database files on the disk is immediately reduced.

Access the Databases tab

Access this view by clicking the appropriate server and selecting the Databases tab.



Alerts associated with the Databases tab

The following alerts are associated with the Databases tab:

Database Full (Percent)

Percentage of the allowable disk space for the database currently used by the sum of data, text, and index. Allowable disk space for the database is calculated by taking into account the current allocated space, auto-growth settings, and available disk space.

Database Read/Write Error Occurred

Failure to read from or write to the hard disk.

Database Status

Multiple types of [Database Status](#) alerts exist.

Index Row hits (Percent)

Number of row hits that part of a clustered index or an entire non-clustered index returns, providing an indication of index effectiveness.

Log Full (Percent)

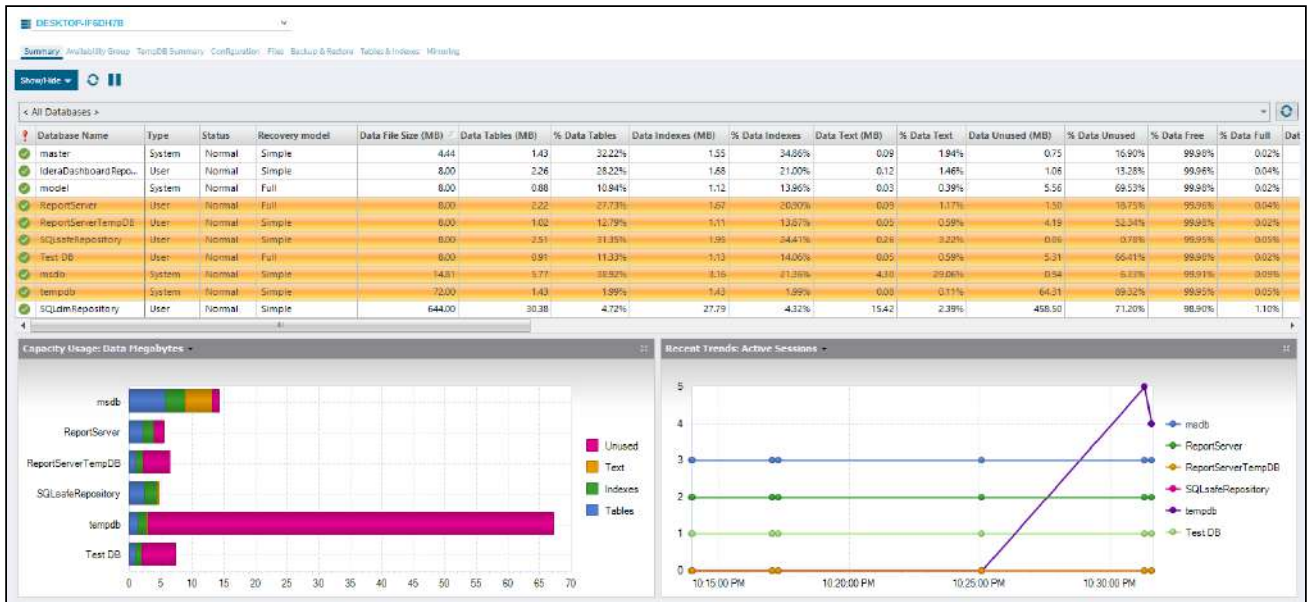
Percentage of the allowable disk space for the database currently used by the transaction log. Allowable disk space for the database is calculated by taking the current allocated space, auto-growth settings, and available disk space.

Table Fragmentation (Percent)

Percentage of pages that are fragmented in each table in the database.

Get the database status summary

The Database Summary tab allows you to view the status of all the databases on the selected SQL Server instance. Along with the database status table, the Summary tab includes charts that provide options for viewing the capacity usage and recent trends of your databases. You can select multiple databases in the list using click+Shift and click+Ctrl to compare several databases.



- ✓ The status column can help you quickly identify databases that have associated alerts and need your Attention.

The Database Summary tab lists each database on the monitored SQL Server instance. The Database Summary tab displays a large variety of database statistics and is an important way to get an overview of the health of your databases. This information gives you a good indication as to how the data is configured in your database, such as comparing the volume of your table data compared to your indexes, or the size of your log files.

Access the Database Summary view

You can open the Summary view of the SQL Diagnostic Manager Databases tab, select the appropriate SQL Server instance, click **Databases**, and select **Summary**.

Database Summary tab charts

The Database Summary tab includes [charts](#) that display capacity usage and recent trends information for your database.

Capacity Usage

You can choose to view how data or the log is used on your database. The drop-down list allows you to choose to view the data or log in megabytes or percentages. For more information, click inside the chart to open the Database Files tab.

Recent Trends

You can choose to view how your database or group of databases are used currently by selecting either active sessions, transactions per second, data size, log size, or log flushes from the drop-down list.

Monitor AlwaysOn Availability Groups

The AlwaysOn Availability Groups view allows you to monitor availability groups, availability databases, and availability replicas.

- ✔ If you have configured the AlwaysOn Availability Groups feature on your SQL 2012 Server, SQL Diagnostic Manager automatically recognizes the availability databases and starts to monitor them.

AlwaysOn Availability Groups are part of an integrated solution, introduced in SQL Server 2012 with the goal of achieving the highest level of data availability and disaster recovery for organizations. Availability Groups grant DBAs the ability to automatically or manually failover a group of databases as a single unit with support for up to four secondary replicas. For additional information on availability groups, see the Microsoft document, [Overview of AlwaysOn Availability Groups \(SQL Server\)](#).

Access the AlwaysOn Availability Groups view

To open the AlwaysOn Availability Groups view click the appropriate monitored SQL Server instance, select **Databases**, and click **Availability Group**.

Database AlwaysOn Availability Groups statistics

In this view, you can see the health of all your availability groups, availability databases and monitored availability replicas. The database table provides the following information for each availability group (as seen in the view by default):

Database Name

Displays the name of the availability database.

Group Name

Displays the name of the availability group for which the connected server instance hosts a replica.

Replica Name

Displays the name of the server instance that hosts the availability replica.

Failover Mode

Displays the failover mode for which the availability replica is configured. The possible failover modes are automatic or manual.

Availability Mode

Describes if data synchronization will use either the synchronous-commit or asynchronous-commit availability mode.

Replica Role

Indicates the current role of the availability replica. Possible values are primary or secondary.

Synchronization Health

Displays database synchronization state in the availability group. Possible values include: not healthy, partially healthy, and healthy.

Redo Queue Size (KB)

Displays the amount of log records from log files in KB that need redoing in the secondary replica to complete synchronization.

Redo Rate (KB/s)

Rate in KB per second at which log records are redone in the secondary database to complete synchronization.

Log Send Queue Size (KB)

Amount of log records in KB needed to ship to the secondary replica to complete synchronization.

Log Send Rate (KB/s)

Indicates the rate in KB per second at which log records are being sent to the secondary replica to complete synchronization.

Listener DNS Name

The network name of the availability group listener.

Listener IP Address

The IP address reserved for the availability group listener. If the nodes of your Windows cluster are on different subnets you will need an IP address for each of those subnets.

Failover Readiness

Indicates whether the secondary database is synchronized with the corresponding primary database. Possible values include: True (the database is marked as synchronized and is ready for a failover) and False (the database is not synchronized and not ready for a failover).

Synchronization Database Status

Indicates whether the availability database is currently synchronized with the primary replica. Possible synchronization states are: not synchronizing, synchronizing, synchronized, reverting, and initializing.

Database Status

Description of the database state of the availability replica. Possible values include: online, restoring, recovering, recovery_pending, suspect, emergency, and offline.

Suspended Status

An AlwaysOn availability database is suspended. Possible values are true or false.

Last Hardened Time

Indicates the time when the log-block identifier was received for the last hardened LSN on the secondary replica.

Operational Status

Indicates the current operational state of the secondary replica. Possible state values include: pending failover, pending, online, offline, failed, failed no quorum, and NULL.

Connection Status

Indicates whether a secondary replica is currently connected to the primary replica. Possible values include: connected and disconnected.

Synchronization Performance (s)

Indicates the time in seconds it takes to synchronize between the primary and secondary replicas.

Estimated Data Loss (s)

Indicates the time difference in seconds of the last transaction log record in the primary replica and secondary replica. If the primary replica fails, all transaction log records within the time window will be lost.

Estimated Recovery Time (s)

Indicates the time in seconds it takes to redo the catch-up time. The catch-up time is the time it will take for the secondary replica to catch up with the primary replica.

Additional columns available through the **Column Chooser:****Database ID**

Identifier of the database, unique within an instance of SQL Server.

FileStream Send Rate (KB/s)

Indicates the rate of the FileStream in KB per second at which transactions are being sent to the secondary replicas.

Group ID

Unique identifier (GUID) of the availability group.

Last Connect Error#

Number of the last connection error.

Last Connect Error Description

Description of the last connection error.

Last Connect Error Time

Timestamp of the last connection error.

Listener Port

The TPC port used by the availability group listener.

Replica ID

Identifier of the availability replica within the availability group.



You can hide and unhide additional columns in the Availability Groups table by right clicking on any header in the grid and selecting a column from the **Column Chooser** dialog.

Availability Groups charts

The AlwaysOn Availability Groups view includes charts that display the queue size and transfer rates of different availability groups.

Queue Size chart

Provides users with graphical details about queues in the AlwaysOn Availability Groups feature. This chart is a stacked bar chart for the "Log Send Queue Size" and "Redo Queue Size."

Transfer Rates chart

Provides users with graphical details about data transfer rates for redo and logs. This chart is a line chart for the "Log Send Rate" and "Redo Rate."

Availability Groups alerts

To view a list of all available alerts, see [Metric alerts](#).

Understand the view colors

The AlwaysOn Availability Groups view uses a color-based alert system. The following colors are associated with a status and action within SQL Diagnostic Manager:

Color	Alert Status	Action
Green	OK	Acceptable threshold where SQL Diagnostic Manager does not generate an alert.
Blue	Informational	Informational threshold where SQL Diagnostic Manager generates an informational alert.
Yellow	Warning	Warning threshold where SQL Diagnostic Manager generates a warning alert.
Red	Critical	Critical threshold where SQL Diagnostic Manager generates a critical alert.

Get the tempdb status summary

The Tempdb Summary tab allows you to view the status of your tempdb database on the selected SQL Server instance. These charts include options for viewing your current capacity usage and recent trends of your files over time. These statistics appear in the Database Statistics report.

The Tempdb Summary tab displays a list of sessions currently using tempdb along with their cumulative usage and tempdb space. These statistics help you get an overview of the health of your tempdb database and gives you a good indication whether you have enough space allocated or whether tempdb issues are causing a performance problem on your server.

Access the Tempdb Summary view

You can open the Tempdb Summary view of the SQL Diagnostic Manager Databases tab, select the appropriate SQL Server instance, click **Databases**, and click Tempdb **Summary**.

Tempdb Summary tab charts

The Tempdb Summary tab includes [charts](#) that display capacity usage and recent trend information for your tempdb database.

Toggle between the Version Store Cleanup and Tempdb Contention charts by using the drop-down list available to the right of the chart title.

Tempdb Space Used by File

You can choose to view how each file is using space on your tempdb database, displayed in megabytes.

Tempdb Space Used Over Time

You can choose to view how your database is used over time based on object type by comparing charts that offer metrics for the Space Used and Space Allocated.

Version Store Cleanup Rate

The [tempdb database version store](#) collects data rows necessary to support snapshot isolation. This chart helps you see the current cleanup rate to avoid filling up tempdb.

Tempdb Contention

The [Tempdb Contention](#) chart displays latch wait time (in milliseconds) for the allocation pages of tempdb. The three tracked allocation page types are:

- Page free space (PFS)
- Global allocation map (GAM)
- Shared global allocation map (SGAM)

Latch contention of this sort is usually an indication that you should create more tempdb data files. In some situations, using Trace Flag 1118 may also alleviate tempdb contention.

Monitor your tempdb database

SQL Diagnostic Manager allows you to monitor your tempdb database with a number of charts and alerts to help you avoid costly performance issues caused by a full tempdb. The tempdb system database is a workspace used to hold temporary user objects, results created through queries and sorts, and other objects created by the SQL Server Database Engine. Because of the large amount of data stored in tempdb, users can run out of disk space, which causes the entire server and all of its databases to become unresponsive. SQL Diagnostic Manager seeks to resolve some of these issues by providing a monitoring solution that features a series of tempdb-specific charts, views, and alerts.

The tempdb system database, along with the Master, Model, and MSDB databases, is provided by default with SQL Server and is a shared resource available to all users connected to a single SQL Server instance. Each time you start SQL Server, it re-creates tempdb based on the Model database. Tempdb can fill up quickly when you are low on disk space or have a low maximum size for database growth. Certain workloads may cause excessive space usage or create contention in tempdb, which can affect performance on the entire server.

✔ Tempdb monitoring is supported on SQL Server 2005 and above servers only.

Tempdb version store

The tempdb version store collects the data necessary to support row versioning. Each time a data value changes, a new version of the row is created and stored for as long as the oldest active transaction needs to access it. Once the row version is no longer needed it is removed from tempdb by a cleanup job which runs once per minute. As a result, long-running transactions prevent cleanup of older entries into the tempdb version store, causing growth which can affect performance and cause tempdb to run out of space.

Tempdb contention

Tempdb resource contention or waits is usually the result of heavy use on too few tempdb files and occurs when the system attempts to allocate pages. The tempdb contention chart displays latch wait time (in milliseconds) for the allocation pages of tempdb. The three tracked allocation page types are:

- Page free space (PFS)
- Global allocation map (GAM)
- Shared global allocation map (SGAM)

Latch contention of this sort is usually an indication that you should create more tempdb data files. In some situations, using Trace Flag 1118 may also alleviate tempdb contention.

Available alerts

Tempdb includes the following specific alerts:

- [Data File Autogrow](#)
- [Log File Autogrow](#)
- [Longest Running Version Store Transaction \(Minutes\)](#)
- [Session Tempdb Space Usage \(MB\)](#)
- [Tempdb Contention \(ms\)](#)
- [Version Store Generation Ratio](#)
- [Version Store Size \(MB\)](#)

Monitor database configuration settings

The Database Configuration view allows you to view the configuration settings of the databases on the selected SQL Server instance. This view contains a single grid that includes the configuration settings for all databases on the selected SQL Server instance.

- ✔ The columns displayed on the Database Configuration view differ between SQL Server 2005 and SQL Server 2008 instances.

You can use the Database Configuration view to analyze configuration differences between databases or databases with incorrect configuration settings.

- ✔ If you experience database issues, the Database Configuration view can help you determine if the problem is caused by configuration errors.

Access the Configuration view

To open the Configuration view of the SQL Diagnostic Manager Databases tab, select the appropriate SQL Server instance, click **Databases**, and select **Configuration**.

Null values in the Collation and Version columns

If the database is not started and AUTO_CLOSE is set to ON, these columns are set to null until the database is started.

View database status

SQL Diagnostic Manager allows multiple simultaneous status indicators for each individual database, very similar to what you see in Microsoft SQL Server Management Studio. You can see the status of a database in the Servers tree by expanding the associated Server tree. The status displayed matches what is shown in Microsoft SQL Server Management Studio in all cases where Management Studio displays a status.

Database alerts and statuses

The following alerts and statuses are associated with Database status:

Default Status Level	Alert/Status	Description
OK	Normal	Everything is okay and the database is operating normally.
OK	Standby	The database is in standby mode.
OK	Read Only	The database is in read-only mode.
OK	Single User	The database is in single-user mode.
OK	Restore Mirror	Restoring the database mirror.
OK	Undetermined	No information is available one this database.
OK	Cleanly Shut Down	The database is shut down and needs no recovery.
OK	DBO Use Only	The database is in a dbo-use mode (restricted_user).
Warning	Restoring	Restoring the database.
Warning	Recovering	Recovering the database.
Warning	Recovery Pending	The database recovery is not yet begun.
Warning	Offline	The database is offline.
Critical	Suspect	The database integrity is suspect.
Critical	Emergency Mode	The database is in emergency mode.

Default Status Level	Alert/Status	Description
Critical	Inaccessible	The SQLdm Monitoring Service is unable to open the database. A serious issue such as a disk failure can cause an inaccessible database.

Change the default alert status level

Alerting is based on the most critical status associated with a database at the time of collection.

- ✔ Note that for certain metrics, using the informational alert means that you no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.

To change the default alert status level:

1. Right-click any server in the Servers tree.
2. Select **Configure Alerts**.
3. Select **Database Status** from the list of alerts.
4. Double-click any of the values on the **Configuration** tab.
5. Edit each alert to associate it with the desired alert status level.

Analyze database files

The Files view allows you to see the file statistics for each of your databases on the selected SQL Server instance. Database capacity and growth are common causes of SQL Server performance bottlenecks.

✔ Database growth statistics are collected every scheduled refresh.

The Files view is an important tool that displays how much disk space you have available to your databases, how much disk space they are using, and the percentage of space used by databases and logs. Database capacity and growth are common causes of SQL Server performance bottlenecks.

For Example: If the % Data Full on a database is over a threshold, it could mean that either the file doesn't have the file auto-growth setting turned on, is about to reach the maximum allowable size, or that the disk the database is on is about to run out of space.

Another common use of the Files view is to identify the fragmentation percentage on an index. Once identified, you can click **Rebuild Indexes** to reduce the fragmentation on the selected index.

Access the Files view

To open the Files view of the SQL Diagnostic Manager Databases tab, select the appropriate SQL Server instance, click **Databases**, and select **Files**.

Get file statistics

Select any database in the list to display the associated file statistics in the Data Files tab of the Files view. Click any file in the list to populate the information displayed in the File Usage chart.

Select the drop-down box above the chart to choose a [chart type](#):

- Current Megabytes
- Current Percent
- Potential Megabytes
- Potential Percent

 You can select multiple databases as well as multiple files to compare file statistics.

Access the Data Files tab

To open the Data Files tab of the SQL Diagnostic Manager Databases Files view, select the appropriate SQL Server instance, click **Databases**, and select **Files**. The Data Files tab is the default view.

Get file group information

The File Groups tab of the Files view displays the selected database. Click a group on the File Groups tab to populate the **File Group Usage** chart. Select the drop-down list above the chart to choose a [chart type](#):

- Current Megabytes
- Current Percent
- Potential Megabytes
- Potential Percent

✔ Select multiple databases and multiple file groups to compare how they are used. This can help you quickly spot issues with your file usage.

Access the File Groups tab

To open the File Groups tab of the SQL Diagnostic Manager Databases Files view, select the appropriate SQL Server instance, click **Databases**, select **Files**, and then click the **File Groups** tab.

View transaction log information

The Transaction Logs tab of the Files view allows you to view the transaction log files for the selected databases. Select the drop-down list above the chart to choose from the following chart types :

- Current Megabytes
- Current Percent
- Potential Megabytes
- Potential Percent

✔ Log files in high transaction rate environments can occupy multiple disks. The Transaction Logs tab allows you to easily monitor these configurations.


Access the Transaction Logs tab

To open the Transaction Logs tab of the SQL Diagnostic Manager Databases Files view, select the appropriate SQL Server instance, click **Databases**, select **Files**, and then click the **Transaction Logs** tab.

Get disk information

The Disks tab of the Files view provides a graphical representation of your current disk usage. Select the drop-down list above the chart to choose from the following chart types:

- Megabytes
- Percent

 You must have [OS metrics enabled](#) to show the information in the charts on the Disks tab.

Access the Disks tab

To open the Disks tab of the SQL Diagnostic Manager Databases Files view, select the appropriate SQL Server instance, click **Databases**, select **Files**, and click the **Disks** tab.

Analyze backups and restores

The Backups & Restores tab allows you to view the backup and restore history of each database on the selected SQL Server instance. You can view the backup and restore history of a database or a group of databases. The history displays in the Backup/Restore History section of the tab. The Backup/Restore section shows the date and time of the backup or restore, the user that initiated it, and the size and path of the backup or restore file.

The Backups and Restores view is a useful way to view the last successful backup, the recovery model used, and the backup history for each of the databases on the SQL Server instance. To see your backup and restore history, click one or multiple databases to populate the Backup and Restore History table.

- ✔ The default filter setting shows a 31-day history. For less frequently run jobs such as quarterly or annual backups, it may be necessary to increase the **Days To Show** setting in the Filters window.

Access the Backups & Restores view

To open the Backups & Restores view of the SQL Diagnostic Manager Databases tab, select the appropriate SQL Server instance, click **Databases**, and select **Backups & Restores**.

Analyze tables and indexes

The Tables & Indexes tab allows you to view statistics for all the tables and indexes on a particular database. To see all the tables and indexes for a database, use the database drop-down list to select your database.

Along with displaying statistics, the Tables & Indexes tab includes the ability to update your statistics and rebuild your indexes on the selected table. These options are available by right-clicking a table and selecting the appropriate option from the context menu. To perform these actions on multiple tables, use the buttons on the ribbon.

- ✔ Table growth statistics are collected on the schedule configured on the Table Statistics tab of the Monitored SQL Server Properties window. This includes the table fragmentation and size statistics.

The Tables and Indexes view provides a good way to view the overall health of your tables. In addition, the Tables and Indexes view allows you to rebuild the index for any of your tables that have a high fragmentation percentage.

- ✔ The Rebuild Indexes function is enabled for User tables. System tables cannot have indexes rebuilt.

Access the Tables & Indexes view

To open the Tables & Indexes view of the SQL Diagnostic Manager Databases tab, select the appropriate SQL Server instance, click **Databases**, and select **Tables & Indexes**.

Use the Table Details section

The Table Details section of the Table and Indexes view allows you to view various details on the selected table. To update the statistics for your tables, click **Update Statistics** at the top of the view. The following tables are available in the Table Details section:

Size

Displays the size of the data, text, and indexes associated with the selected table.

Dependencies

Lists other database objects that the selected table contains references to and other database objects that reference the selected table. In addition to the object name, the type of object and action associated is listed.

Indexes

Lists the associated indexes and index information for the selected table.

Index Statistics

Lists the number of columns, average length, average row hits, and row hits percentage for each column in an index associated with the selected table. It also lists the data distribution data values and occurrences of each value.

Table Size Details chart

The Table Size Details chart displays the size of each component in a table. To view the Table Size Details chart, select a table from the list and click the Size tab in the Table Details section.

Analyze table dependencies


The Table Dependencies tab displays the objects the selected table references and is referenced by, the type of dependency, and the action of the dependency. This allows you to quickly see how the table is utilized and how it is affected by and affects other tables and objects.

Indexes tab

The Indexes tab displays all indexes defined for the selected table and an overview of each index.

Analyze index statistics

The Tables and Indexes view allows you to see general table and index statistics for the database selected. You can also rebuild indexes for highly fragmented tables and update your statistics, which are displayed in the chart and tables in the Table Details section of the view. For more information, see the Microsoft document, [Tables and Indexes](#).

 The Collection Service may not gather fragmentation data for a particular table because:

- Table statistics collection has not occurred
- The database hosting the table was excluded from collection
- The table size does not meet the minimum size threshold
- The database hosting the table is not accessible
- The table is locked

Update table statistics

The Update Statistics option on the Tables and Indexes view of the Databases tab allows you to update your Table Statistics information. When you click **Update Statistics** from the SQL Diagnostic Manager Console, the Select Tables window opens and allows you to choose the tables to update.

Rebuild indexes

The Rebuild Indexes option on the Tables and Indexes view allows you to select tables to rebuild. Rebuilding your tables allows you to decrease the amount of fragmentation on a table. High table fragmentation can lead to overall performance bottlenecks on your SQL Server instance.

When you click **Rebuild Indexes** from the SQL Diagnostic Manager Console, the **Select Tables** window opens and allows you to choose the table indexes to rebuild.

- ✔ If you are rebuilding indexes on a very large table the rebuild process could take several minutes to complete. While the rebuild is in process, the table is locked and users are unable to access it.

Once the indexes are rebuilt, the **% Fragmentation (Current)** column displays showing you the results.

- ✔ If you are rebuilding a very large index, the SQL Diagnostic Manager Console checks for the rebuild status for 90 seconds. If the rebuild is not complete after 90 seconds, the **% Fragmentation (Current)** column does not display the new value until the next table statistics collection occurs.

Monitor database mirroring

The Database Mirroring view allows you to monitor the mirrored databases set up on monitored SQL Server instances. You can view the mirrored databases by clicking **Mirroring** on the Databases tab.

- ✔ If you have enabled database mirroring on your SQL Server, the SQL Diagnostic Manager Database Mirror Monitoring view is enabled by Default.

Database mirroring is a process available in SQL Server 2005 and 2008 that moves database transactions from one SQL Server database to another SQL Server database on a different instance of SQL Server.

Database mirroring allows you to configure a duplicate of your important databases for availability and redundancy.

Access the Mirroring view

You can open the Mirroring view of the SQL Diagnostic Manager Databases tab by selecting the appropriate SQL Server instance, and clicking **Databases > Mirroring**.

Database mirroring statistics

In this view, you can see all the mirrored databases on all the monitored SQL Server instances, their current status, and history. The database table provides the following information:

Database Name

The name of the database.

Server Instance

The SQL Server instance hosting the mirrored database.

Current Role

Describes whether the database is the principal or the mirrored database.

Partner Instance

The name of the SQL Server instance that is the mirror partner.

Mirroring State

The status of the database mirror, color-coded based on the configured alert level.

Witness Connection

The status of the connection, color-coded based on the configured alert level.

Operational State

Describes whether the database mirroring session is operating in a preferred configuration based on how the user established which SQL Server instance hosts the principal and which SQL Server instance hosts the mirrored database.

Operating Mode (Hidden)

Shows the current operating mode: High availability with automatic fail-over (has a witness, synchronous); High availability without automatic fail-over (no witness, synchronous); High performance (no witness, asynchronous).

Available details for a specific session

Select a session, and then click the Status tab to view the following details. You can also view historical information for a selected session using the History tab. This information includes the history of the metrics over time for both the local and remote partner.

Unsent log (in kilobytes)

Size of the log waiting in the send queue on the principal, displayed in kilobytes.

Oldest unsent transaction (dd.hh:mm)

Date and time of the oldest unsent transaction in the send queue on the principal.

Time to send log (estimated)

Estimated number of minutes required by the principal to send the log waiting in the send queue to the mirror based on the current send rate. Note that this rate is affected by the rate of incoming transactions.

Send rate (in kilobytes per second)

Rate at which the principal is sending transactions to the mirror, displayed in kilobytes per second.

Rate of new transactions (in kilobytes per second)

Rate at which transactions are added to the unsent queue on the principal.

Unrestored log (in kilobytes)

Size of the log waiting in the redo queue on the mirror, in kilobytes.

Time to restore log (estimated)

Estimated number of minutes required to apply the log in the redo queue to the mirror database.

Current restore rate (in kilobytes per second)

Rate at which transactions in the redo queue are applied to the mirror database, in kilobytes per second.

Mirror commit overhead

Size of the log waiting in the redo queue on the mirror, in kilobytes. This delay is due to mirror operations and is the first indication that the traffic exceeds the capabilities of the hardware.

Time to send and restore all current log (estimated)

Estimated number of minutes required to process all outstanding transactions on both the principal and mirror.

Operating mode

Displays one of the following operating modes for the current database mirroring session: High availability with automatic fail-over (has a witness, synchronous), High availability without automatic fail-over (no witness, synchronous), or High performance (no witness, asynchronous).

Witness Address

Network address of the witness SQL Server instance.

Mirror Address

Network address of the mirror SQL Server instance.

Principal Address

Network address of the principal SQL Server instance.

Available actions in the Database Mirror Monitoring view

Each row of the mirrored databases table provides a menu of actions you can perform:

Failover

Select **Fail over to partner** to fail over the principal role between the two partner databases in the mirroring session.

The duration of a mirroring failover operation varies according to the amount of log in the redo queue on the mirror. All transactions in the redo queue must roll forward before the mirror can assume the role of Principal. After the failover is initiated, the Principal changes to PENDING_FAILOVER while the mirror synchronizes. Once complete, there is a brief period where user connections are terminated and while roles are switching the mirroring status is undetermined. If a refresh occurs at this time, your mirrored database does not show in the Mirrored Databases grid. This lasts for a moment and your next refresh reflects the switched roles.

For more information about the failover function, see [SQL Server Books Online](#).

Suspend or Resume Monitoring

Use **Suspend/resume session** to pause all mirroring operations. While mirroring is paused in "High Availability" mode, all transactions opened against the principal are queued for transmission to the mirror partner once the session resumes.

Set mirror relationship as "Failed over" or "Normal"

Select **Mark session as "failed over"** or **Mark session as "normal" operational status** to define the status of a mirror relationship. Once you have set a preferred operational state, SQL Diagnostic Manager alerts you if the configuration differs from your selection.

Clear the role preference for this mirroring session

Use **Clear preferred role for this session** to clear the preferred role for the session and stop receiving alerts for differences between the current configuration and the Normal status.

Available alerts for mirrored databases

You can [configure the following alerts for mirrored databases](#) .

Mirror Commit Overhead

Average delay (in milliseconds) on the principal server instance for each transaction due to database mirroring.

Mirrored Server Role Change

A change in role occurred on a mirrored database on the monitored server.

Mirroring Oldest Unsent Transactions

The age of the oldest unsent transaction in minutes on the principal. This is only meaningful on the principal server instance.

Mirroring Preferred Configuration

A change occurred in terms of the preferred operational status.

Mirror Status

Status of the mirrored database such as synchronized, suspended or disconnected.

Mirroring Unrestored Log

Size of the redo queue on the mirror in kilobytes. The redo queue is the size in kilobytes of the outstanding transactions on the mirror that are not yet applied.

Mirroring Unsent Log

Size of the unsent log in the send queue on the principal in kilobytes.

Mirroring Witness Connection

If a witness is configured, this alert displays whether the witness is connected.

Monitor services

The Services tab allows you to view the status and manage your SQL Server services including the following services:

- SQL Server Agent
- SQL Server Browser
- Full-Text Search (Note that SQL Server 2008 no longer has a separate full-text search service. If the monitored instance is SQL Server 2008, the full text search does not appear on the grid or graph.)
- DTC (Distributed Transaction Coordinator service)
- Replication services

- ✓ The Start and Stop service actions are not supported for virtual SQL Server instances or instances located on a Windows cluster node. To start or stop a service on a virtual SQL Server 2008 instance, use the Microsoft Failover Cluster Management tool. For SQL Server 2005 or earlier, use the Microsoft Cluster Administrator tool to manage services.

Access the Services tab

Access this tab by selecting the appropriate monitored server and clicking the Services tab.

The screenshot shows the IDERA SQL Diagnostic Manager (SQLDM) interface. The 'SERVICES' tab is active, displaying a list of services for the server 'DESKTOP-IF8QHTB'. The services listed are:

Service	Status	Running Since	Service Name	Startup Type	Log On As	Description
Distributed Transaction Coordinator	Stopped		MSDTC	Manual	NT AUTHORITY\NETWORKSERVICE	Coordinates transactions that span multiple resource managers, such as databases, message queues, and file systems.
SQL Server	Running	6/2/2020 12:28:01 PM	MSSQLSERVER	Auto	NT SERVICE\MSSQLSERVER	Provides storage, processing and controlled access of data, and rapid transaction processing.
SQL Server Agent	Running	6/2/2020 12:28:01 PM	SQLSERVERAGENT	Manual	NT SERVICE\SQLSERVERAGENT	Executes jobs, monitors SQL Server, fires alerts, and allows automation of some administrative tasks.
SQL Server Browser	Stopped		SQLBROWSER	Disabled	NT AUTHORITY\LOCALSERVICE	Provides SQL Server connection information to client computers.

Below the table is a 'Service Availability' graph showing the status of services over time. The graph displays four bars representing the status of the services: Distributed Transaction Coordinator (red, Stopped), SQL Server (green, Running), SQL Server Agent (green, Running), and SQL Server Browser (red, Stopped). The x-axis shows time from 4:04:40 PM to 4:05:25 PM. A legend on the right indicates the status colors: Running (green), Stopped (red), Unavailable (yellow), Not installed (grey), and Unable To Monitor (blue).

Alerts associated with the Services tab

The following alerts are associated with the Services tab:

Cluster Active Node

The active node of the cluster is a non-preferred node. You can configure the preferred node under **Server Properties > Cluster Settings**.

Cluster Failover

There is a change in the active node. This alert remains active for the amount of time specified in the advanced settings for that alert.

Distribution Latency (Seconds)

Time in seconds that a replication transaction is holding at the Publisher and waiting for a Distributor.

DTC Status

Displays the status of the Distributed Transaction Coordinator service.

Full-Text Search Status

Displays the status of the Full-Text Search service.

Last Full-Text Catalog Update (Hours)

The number of hours beyond which the Full-Text Catalog statistics data is considered outdated.

Latent replication transaction (Count)

The number of transactions not written to the Distribution database.

SQL Server Agent Job Completion

Provides the status of the Agent Job Completion metric. Values for this alert include **Succeeded**, **Retry**, **Canceled**, **Failed**, and **Unknown**.

SQL Server Agent Job Failure

Indicates that a scheduled job processed by the SQL Server Agent service ended abnormally.

SQL Server Agent Long Running Job (Minutes)

Provides the maximum number of minutes used to complete a scheduled SQL Server Agent job.

SQL Server Agent Long Running Job (Percentage)

Indicates the percentage of time a scheduled job is using over what it normally uses to complete the job.

SQL Server Agent Status

Displays the status of the SQL Server Agent service.

SQL Server Active Directory Helper Service Status

Displays the status of the SQL Server Active Directory Helper service.

SQL Server Browser Service Status

Displays the status of the SQL Server Browser service.

SQL Server Status

Displays the status of the SQL Server service.

Unsubscribed Transactions (Count)

The number of non-subscribed transactions written to the Distribution database.

Unsubscribed Transactions (Seconds)

The number of seconds a transaction written to the database waits before subscription.

Get an overview of your services performance

The Services Summary tab includes the status of your Services. You can start and stop your Distributed Transaction Coordinator (DTC), or Full-Text Search by selecting the appropriate button.

- ✔ SQL Server 2008 and above no longer have a separate full-text search service. If the monitored instance is SQL Server 2008 or above, the full text search does not show on the grid or graph.

The [Service Availability chart](#) displays the DTC, Full-Text Search, SQL Server, SQL Server Browser, and SQL Server Agent status.

- ✔ An Unavailable status in the chart is caused by a pending status change such as starting, stopping, pausing, and continuing.

Access the Services Summary view

To open the Summary view of the SQL Diagnostic Manager Services tab, select the appropriate SQL Server instance, click **Services**, and select **Summary**.

Monitor SQL Agent jobs

The SQL Agent Jobs view allows you to view all your SQL Agent Jobs for the selected SQL Server instance. You can view the Job History of SQL Agent Jobs in the Job History list and Job Steps by clicking the **+** button next to the **Job Name**.

- ✔ SQL Server 2005 users may notice that SQL Diagnostic Manager does not display the Job History for the listed SQL Agent jobs. This issue is the result of a SQL Server 2005 security setting. You can override this security setting by using the following statement on the target server:

```
sp_configure 'agent XPs',1  
reconfigure with override
```

Access the SQL Agent Jobs view

To open the SQL Agent Jobs view of the SQL Diagnostic Manager Services tab, select the appropriate SQL Server instance, click **Services**, and **SQL Agent Jobs**.

Last Run Outcome colors

The Last Run Outcome column of the SQL Agent Jobs page is color coded to quickly alert you of issues needing your attention. The colors used in this column depend on the use of the SQL Agent Job Completion metric in the Alert Configuration dialog.

If this metric is enabled, the Last Run Outcome column is colored based on the correlation of the outcome of the job or job step and the state that outcome corresponds to in the SQL Agent Job Completion metric. If this metric is not enabled, then SQL Diagnostic Manager displays results in red if the outcome is Failed or Canceled, and green if the outcome is Succeeded. All other outcome states remain the default color.

To customize the Last Run Outcome settings for your SQL Agent jobs:

1. Right-click a SQL Server instance in the Servers tree.
2. Select **Configure Alerts**.
3. Select the **SQL Server Agent Job Completion** metric.
4. Click the **Edit** button.
5. Make the necessary changes to customize your alert states.
6. Click **OK**.

View message information about a specific job

Use the Job History list to see more information about a specific job. You can review the full message by right-clicking the appropriate job name, and then selecting **View Message**. A copy feature allows you to cut and paste the message detail into another format for additional use.

Monitor full text searches

Use the Full-Text Search to monitor and diagnose problems with the SQL Server Full-Text Search service. The view is composed of a single list that displays information about each Full-Text catalog maintained on a monitored SQL Server. You can view Catalog details by selecting a Full-Text Catalog from the list.

The Full-Text Search view lists all the full-text search catalogs that are hosted on the SQL Server instance as well as their statuses. You can also start or stop the Full-Text Search service, optimize or rebuild the catalogs.

For each catalog listed, the database, status, size, and last population date is listed. In addition, selecting any catalog from the list populates the **Catalog Details** section. This section includes information on the tables, rows, columns, catalogs, language, and column data types included in the Full-Text search.

Access the Full-Text Search view

To open the Full-Text Search view of the SQL Diagnostic Manager Services tab, select the appropriate SQL Server instance, click **Services**, and select **Full-Text Search**.

Optimize or Rebuild my Full-Text Catalog

Optimize


When you select the **Optimize** button, SQL Server optimizes the space utilization of the catalog and improve query performance as well as the accuracy of relevance rankings of search results.

Rebuild

When you select the **Rebuild** button, SQL Server repopulates the full-text search index.

Analyze replication information

The Replication view allows you to analyze data replication details for the selected SQL Server instance. Included are the topology and key metrics that detail the current state of active subscribed and non-subscribed transactions on all SQL Server instances related to replication on the monitored SQL Server. For more information about replication, see the Microsoft document, [SQL Server Replication](#).

 **Collecting replication statistics** can have a performance impact on the monitored SQL Server instance.

Using the Replication view

The following elements in the Replication view help you make sure that proper replication is occurring and that your data is distributed to the correct subscribers.

Replication Trends

The Replication Trends view on the Services tab, displays information about the following metrics:


- The **Non-Distributed Count** graph tracks the number of replication transactions published by the Publisher but not received by the Distributor.
- The **Non-Distributed Time** graph tracks the time in seconds that a replication transaction has been holding at the Publisher and waiting to be received by a Distributor.
- The **Non-Subscribed Count** graph tracks the number of replication transactions received by the Distributor and not received by a Subscriber.
- The **Non-Subscribed Time** graph tracks the time in seconds that a replication transaction has been holding at the Distributor and waiting to be received by a Subscriber.

Replication Topology

The Replication topology grid displays all replication sessions in which the selected server is participating, whether it is a publisher, distributor, or subscriber. For merge and snapshot replication, SQL Diagnostic Manager must monitor the distributor. For transactional replication, we recommend that you monitor the distributor, publisher, and subscriber for a complete set of metrics. Note that SQL Diagnostic Manager displays a subset of data for each monitored participant. Right-click a row or session to navigate to the participant for more information.

Detailed Overview

The Detailed Overview tab provides an overview of transactional replication specific to the publisher, distributor, and subscriber. In addition, SQL Diagnostic Manager also displays an overview of the selected merge replication session gathered from the distributor.

 Times displayed in the subscriber grid are local to the subscriber.

Non-Subscribed Queue

The Non-Subscribed Queue tab contains the Entry Date, Subscription Database, Wait Time, and Command contained in the queue of active non-subscribed commands. Once a command exceeds the retention period, SQL Diagnostic Manager switches the command to inactive and no longer displays the detail.

Access the Replication view

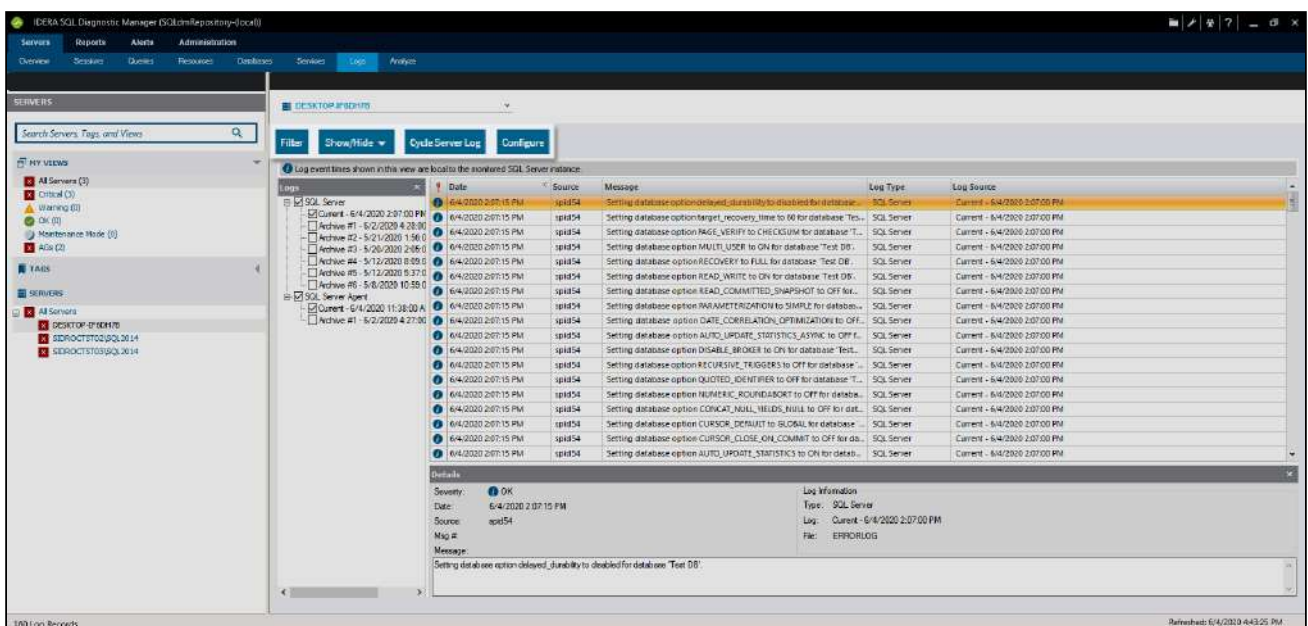
To open the Replication view of the SQL Diagnostic Manager Services tab, select the appropriate SQL Server instance, select **Services**, and click **Replication**.

Monitor logs

The Logs tab allows you view SQL Server related logs. Use these logs to diagnose SQL Server problems where a history of events is necessary to form context. The current log is selected by default but you can select archived or even multiple archived logs to view or search. Filter and Search options allow you to locate the areas of the logs that are most important.

Access the Logs tab

Access this tab by clicking the appropriate monitored server and selecting **Servers>Logs**.



Available actions

Actions

Click **Cycle Server Log** to archive the information in the most recent log and start a new log.

Click **Configure** to determine the number of SQL Server logs retained by SQL Server.

Details section

The Details section provides more information for the selected event.

Filter

The Filter options allow you to return specific data. Note that these selections work only once you enable log alerts.

Logs tab

The Logs tab allows you to look at all the events taking place on your SQL Server instance. Each event, when selected, contains detailed information in the Details section of the tab which allows you to diagnose specific issues that occur.

The archive number is the file number associated with the log in SQL Server while the date is the last modified date of the file on disk. If the archive numbers seem out of order, check the error log.

For Example : If one of your SQL Server instances restarts, the event log can tell you the time of the restart. This helps you diagnose issues that occur on your SQL Server instances.

Filter SQL Server events

The Filter Settings window allows you to select:

- The time range to filter events, such as **Date Range-Begin** and **Date Range-End**.
- The source of the log entry, such as the session ID number.
- The message severity, such as **OK**, **Warning**, or **Critical**.

If the alert is disabled, all messages are informational.

Access the Filter Settings window

To open the Filter Settings window from the SQL Diagnostic Manager Logs tab, select the appropriate SQL Server instance, click **Logs**, and select **Filter**.

Settings Server Log filters

To Filter the Server Log:

1. Click the **Filter** button on the Logs tab.
2. Verify that SQL Diagnostic Manager displays the correct **Source**.
3. Make the necessary changes in the **Message Severity** area.
4. Type a **Date Range-Begin** and **Date Range-End** in the **Period** section of the Filter Settings window.
5. Click **OK**.

Search for SQL Server events

The **Find** window, accessed by clicking **Search**, allows you search for text in the event messages.

To search for text in your Server logs:

1. Click **Search** on the Logs tab.
2. Enter the search term in the **Find what** field.
3. Select the field you want to search (**Message, Msg #, or Source**) from the **Look in** drop-down list.
4. Select your **Find** options.
5. Click **Find Next**.
6. Keep clicking **Find Next** to scroll through the matches to locate the specific event.

Cycle the SQL Server log

Click the **Cycle Server Log** button to archive what is in the most recent (current) log and start a new log.

To cycle the Server Logs:

1. Click **Cycle Server Log**.
2. Select the log types you want to cycle (SQL Server or SQL Server Agent).
3. Click **OK** to cycle the logs.

Configure logs

The Configure Logs window allows you to determine the number of SQL Server Logs retained by SQL Server. You can access this window by clicking **Configure** on the Logs tab.

Check the **Unlimited** check box to retain all **Event Logs** or enter the number to keep only a number of the most current logs. SQL Diagnostic Manager deletes older logs.

- ✔ The number indicated on the Configure Server Logs window changes the number of SQL Server Logs stored by SQL Server, but not the number of SQL Server Agent logs stored.

Analyze your SQL Server instance

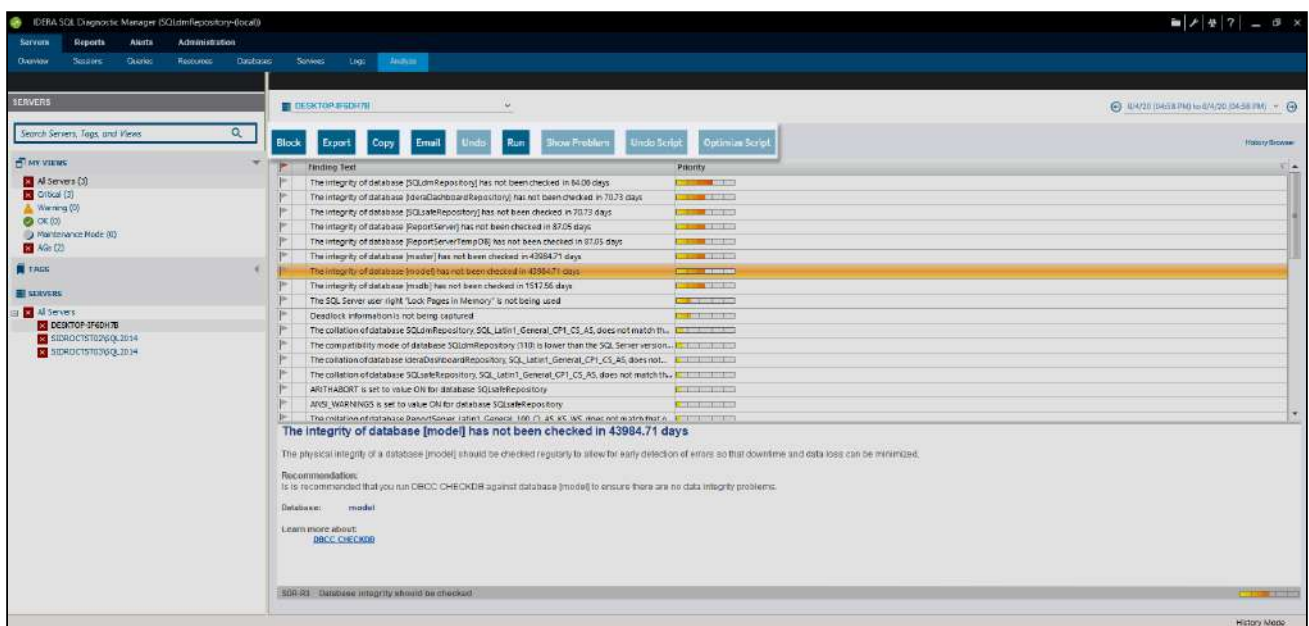
In SQL Diagnostic Manager you can run a prescriptive analysis on a specific SQL Server instance to identify and resolve SQL Server performance problems. The analysis engine scans your SQL Server configuration for potential problems and the health of your databases, resulting in a useful set of recommendations for improving performance.

Prescriptive analysis targets some of the most common areas of SQL Server performance problems, such as queries, server configuration, security, database objects, memory, and more. Its easy to work with the interface, it makes this feature accessible to a broad range of users such as the less experienced DBA or the developer in need of testing database performance.

 SQL Diagnostic Manager now includes new recommendations for Azure and Amazon RDS instances.


Access the Analyze tab

Access this tab by clicking the appropriate monitored server and selecting the Analyze tab.



In the **Analyze** tab you can:

- Run a prescriptive analysis on your SQL Server on-demand or schedule a regular analysis of your SQL Server instance.
- Fine-tune your analysis.
- Identify a problem and optimize scripts.
- View your prescriptive analysis results.

 Prescriptive analysis uses the best method to collect query data according to the SQL Server version. If your monitored server is SQL Server 2008 or a later version, Extended Events is used in place of SQL Trace.

- ✔ If you do not wish to run a full analysis of your SQL Server, you can diagnose individual queries. For more information, see [Diagnose queries](#).

Run a prescriptive analysis on your SQL Server

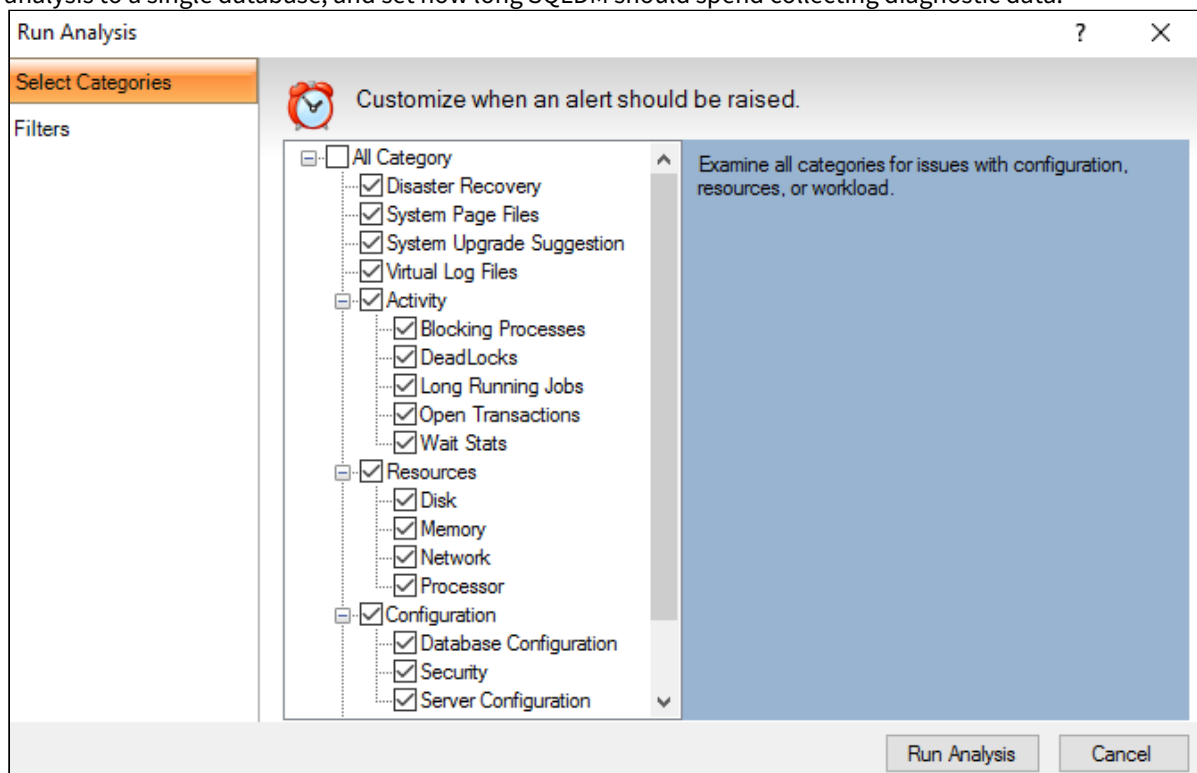
When configuring your analysis, you can select any combination of [performance categories](#) to focus the analysis. Each category contains a set of performance metrics and their related recommendations. By picking a category, you are identifying which SQL Server areas concern you most.

Based on the chosen categories, SQL Diagnostic Manager polls your target SQL Server instance to gather the appropriate metrics and determine which recommendations match the current performance issues.





To run an analysis:

1. Click **Run Analysis** in the Analyze tab.
2. When the Run Analysis window opens, you can select all the categories depending on what you want to include in your analysis. In the Filters tab, you can include specific applications (% wildcard), limit your analysis to a single database, and set how long SQLDM should spend collecting diagnostic data.




3. Once you finished with the Analysis configuration, click **Run Analysis** to start with the Analysis, or click **Cancel** if you want to cancel the Analysis.
4. When the analysis is complete, SQL Diagnostic Manager lists all the performed analysis results by date.
5. Double-click any of the Analysis to review the priority-ranked list of recommendations.

When you run the optimize script, a  displays next to the optimized recommendation.

-  You can also copy the script to a Notepad or any other text editor to examine the code and research other options further before making the change.

Undo Script

The **Undo Optimization** window displays the script that is used to enact the recommended change. Review the script and either click the **Run** button to initiate the undo or click **Cancel** to return to the recommendation you were viewing. The undo script can be copied to the clipboard and used at a later time.

-  Executing the undo script on a recommendation that is not already optimized can harm your SQL Server environment. Before executing the undo script, be sure that the corresponding optimization is performed.

Show Problem

The **Show Problem** button launches the SQL Viewer window with the SQL text associated with the recommendation highlighted.

Additional actions

Block

Block a particular recommendation or recommendations to fine-tune your analysis.

Export

Export your recommendations in .xls file format.

Copy

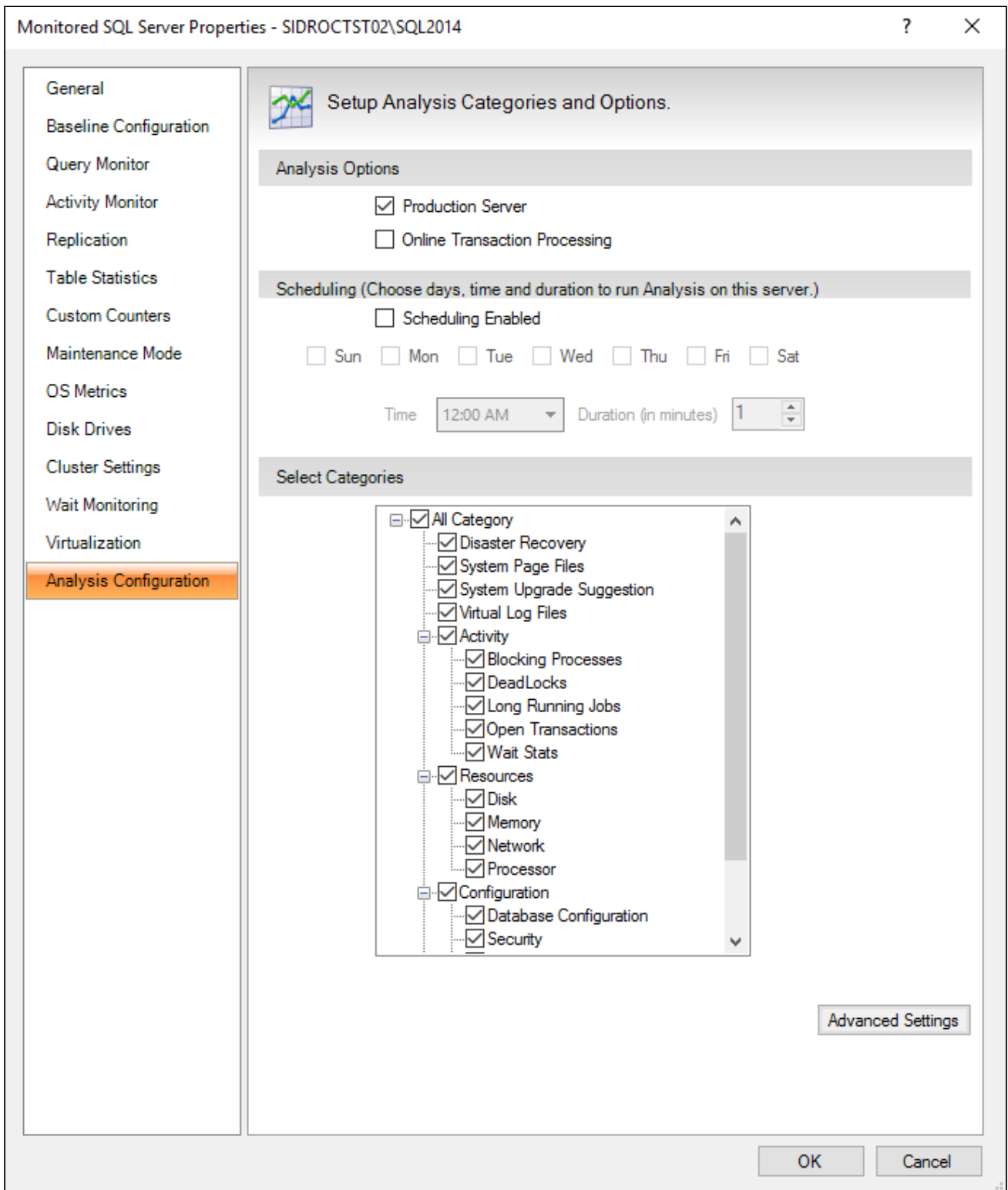
Copy your recommendations to the clipboard so that you can view the text in a text editor.

Email

Email your analysis results with recommendations.

Configure prescriptive analysis options

In the Analysis Configuration tab of the Monitored SQL Server Properties window you can schedule a prescriptive analysis of your SQL Server instance and configure your analysis options.



Access the Analysis Configuration tab

To can access the Analysis Configuration tab of the Monitor SQL Server Properties window, right-click the appropriate monitored SQL Server instance, select **Properties**, and click **Analysis Configuration**.

Specify analysis options

SQL Diagnostic Manager provides two analysis options: Production server or Online Transaction Processing (OLTP).


- Check **Production Server**, if your instance is being used as a production server.
- Check **Online Transaction Processing** (OLTP), if your instance has OLTP enabled.

Schedule an analysis

SQL Diagnostic Manager allows you to schedule the days and time for a *regular prescriptive analysis* and/or specify the duration of a *workload analysis* on your SQL Server instance. Scheduled analyses are an important way to track the health of your SQL Server environment. For more information on this feature, see [Schedule analysis](#).

Select categories

When configuring your analysis, you can select any combination of performance categories to focus the analysis. Each category contains a set of performance metrics and their related recommendations. By picking a category, you are identifying which SQL Server areas concern you most. For additional information on the different categories, see [Performance categories](#).

 The more categories you pick, the longer the analysis needs to run to ensure all relevant data has been collected and thoroughly analyzed.

Using advanced analysis configuration options


You can choose to block recommendations and databases from the analysis, as well as apply additional filters using the options available on the Server Settings window. To access this window, click the [Advanced Settings](#) button on the Analysis Configuration tab.

Performance categories

On the Select categories section, you can select one or more of the following performance categories for use in the detailed analysis:

General

Disaster Recovery, System Page File, System Upgrade Suggestion, and Virtual Log Files.

 The System Page Files, System Upgrade Suggestion, and Virtual Log Files performance categories are not available in SQL Diagnostic Manager 10.0.

Activity

Blocking Processes

Detects any processes that are blocking other processes.

Deadlocks

Detects any deadlocks on the SQL Server instance.

Long Running Jobs

Identifies any long-running jobs.

Open Transactions

Identifies any long-running open transactions and determines whether the associated T-SQL code needs corrections to improve performance.

Wait Stats

Identifies any waits that are affecting SQL Server performance.

Resources

Disk

Detects whether disk settings and associated problems are affecting SQL Server performance.

Memory

Examines the server memory settings and determines whether they affect performance.

Network

Detects network problems or configuration settings that affect SQL Server performance.

Processor

Detects processor problems and associated configuration settings.

Configuration

Database Configuration

Identifies which database configuration settings improve SQL Server performance.

Security

Identifies non-optimal security settings on the SQL Server instance.

Server Configuration

Detects server configuration settings that can decrease the performance of your SQL Server instance.


Workload

Index Optimization

Determines whether index settings or problems with the indexes themselves are negatively affecting SQL Server performance.

Query Optimization

Detects query problems that negatively affect your SQL Server performance.

 The Index Optimization and Query Optimization categories return [heavy recommendations](#) if you run a workload analysis and they return [light recommendations](#) if you run a regular analysis.

Index and Query Optimization performance categories

The Index Optimization and Query Optimization categories return light recommendations if you run a regular analysis and they return heavy recommendations if you run a workload analysis.

Lightweight recommendations

Index Optimization

ID	Recommendation
SDR-I1	AutoStatsRecommendation
SDR-I10	IndexUnderutilizedRecommendation
SDR-I16	IndexUnusedRecommendation
SDR-I17	IndexPageLatchContentionRecommendation
SDR-I2	FragmentedIndexRecommendation
SDR-I21	OverlappingIndexRecommendation
SDR-I22	NonClusteredMatchingClusteredIndexRecommendation
SDR-I23	NonIncrementalColumnStatOnPartitionedTable
SDR-I24	HighModificationsSinceLastStatUpdate
SDR-I25	HashIndexFewBuckets
SDR-I26	HashIndexLargeDuplicateKey
SDR-I27	ScannedHashIndex
SDR-I28	HashIndexTooManyBuckets
SDR-I29	RarelyUsedIndexOnInMemoryTable
SDR-I3	OutOfDateStatsRecommendation
SDR-I30	ColumnStoreIndexMissingOnLargeTables

ID	Recommendation
SDR-I31	FilteredColumnNotInKeyOfFilteredIndex
SDR-I5	DisabledIndexRecommendation
SDR-I6	DuplicateIndexRecommendation
SDR-I7	HypotheticalIndexRecommendation
SDR-I8	PartialDuplicateIndexRecommendation
SDR-I9	IndexLockContentionRecommendation

Query Optimization

ID	Recommendation
SDR-Q3	SQLModuleOptionRecommendation
SDR-Q37	Flag4199LowCompatible
SDR-Q38	Flag4199AllDbCompatible
SDR-Q39	QueryStoreDisabled
SDR-Q40	QueryStoreOutOfSpace
SDR-Q41	QueryStoreAlmostFull
SDR-Q42	PlanGuidesUsedOverQueryStore
SDR-Q43	FrequentlyExecutedProcedureWithHighCPUTime
SDR-Q44	NewCardinalityEstimatorNotBeingUsed
SDR-Q46	Top10QueriesWithLongestAverageExecutionTime
SDR-Q47	Top10QueriesConsumingMostIO
SDR-Q48	QueriesWithDoubleIncreaseInExecutionTime
SDR-Q49	Top10QueriesHavingLongerDurationInLastHour

ID	Recommendation
SDR-Q50	QueriesWithFourDifferentPlanInTwoDays

Heavy weight recommendations

Index Optimization

ID	Recommendation
SDR-I4	MissingIndexRecommendation
SDR-I19	NoColumnStatsRecommendation

Query Optimization

ID	Recommendation
SDR-Q11	TSqlRecommendationWithTable(UnfilteredDelete)
SDR-Q13	TSqlHintRecommendation(HintAbuse)
SDR-Q16	TSqlRecommendation(LeftExpressionInWhereClause)
SDR-Q17	TSqlRecommendation(LikeUseNotNeeded)
SDR-Q18	TSqlNestedMinMaxRecommendation(NestedMinMax)
SDR-Q19	NoJoinPredicateRecommendation
SDR-Q2	TSqlCursorRecommendation(FastForwardCursor)
SDR-Q20	TSqlRecommendationWithColumn(NotInUsedOnNullableColumn)
SDR-Q22	TSqlRecommendation(SelectStarAbuse)
SDR-Q24	TSqlRecommendation(FunctionInWhereClause)
SDR-Q26	TSqlRecommendation(UnionSetAbuse)
SDR-Q28	TSqlRecommendation(TopVsRowCount)
SDR-Q29	TSqlRecommendation(AnsiTopEquivalent)


ID	Recommendation
SDR-Q34	TSqlRecommendation(NakedInsert)
SDR-Q35	UserDefinedFunctionInPredicateRecommendation
SDR-Q35	IntrinsicFunctionInPredicateRecommendation
SDR-Q35	ImplicitConversionInPredicateRecommendation
SDR-Q6	CachedPlanMisuseRecommendation
SDR-Q8	TSqlRecommendation(TwoExpressionCoalesce)

Schedule analysis


SQL Diagnostic Manager allows you to schedule the days and time for a *regular prescriptive analysis* on your SQL Server instance. Scheduled analyses are an important way to track the health of your SQL Server environment. To get the most out of this feature, IDERA recommends that you schedule the analysis to run during busy periods or during times your SQL Server users are experiencing problems.

To schedule an analysis:

1. Right-click the appropriate monitored SQL Server instance and select **Properties > Analysis Configuration**.
2. On the Setup Analysis Categories and Options window, under the Scheduling section, click the **Scheduling Enabled** check box.
3. Select the days and time you want the analysis to start. For best results, select a time when the SQL Server instance is busy.

 The specified days and time only apply to prescriptive analysis. Workload analysis is not impacted by this feature.

4. Specify the total **Duration** (in minutes) that you want SQL Diagnostic Manager to take collecting and analyzing data from this server.

 The specified duration only applies to workload analysis. Prescriptive analysis is not impacted by this feature.

5. Click **OK**.

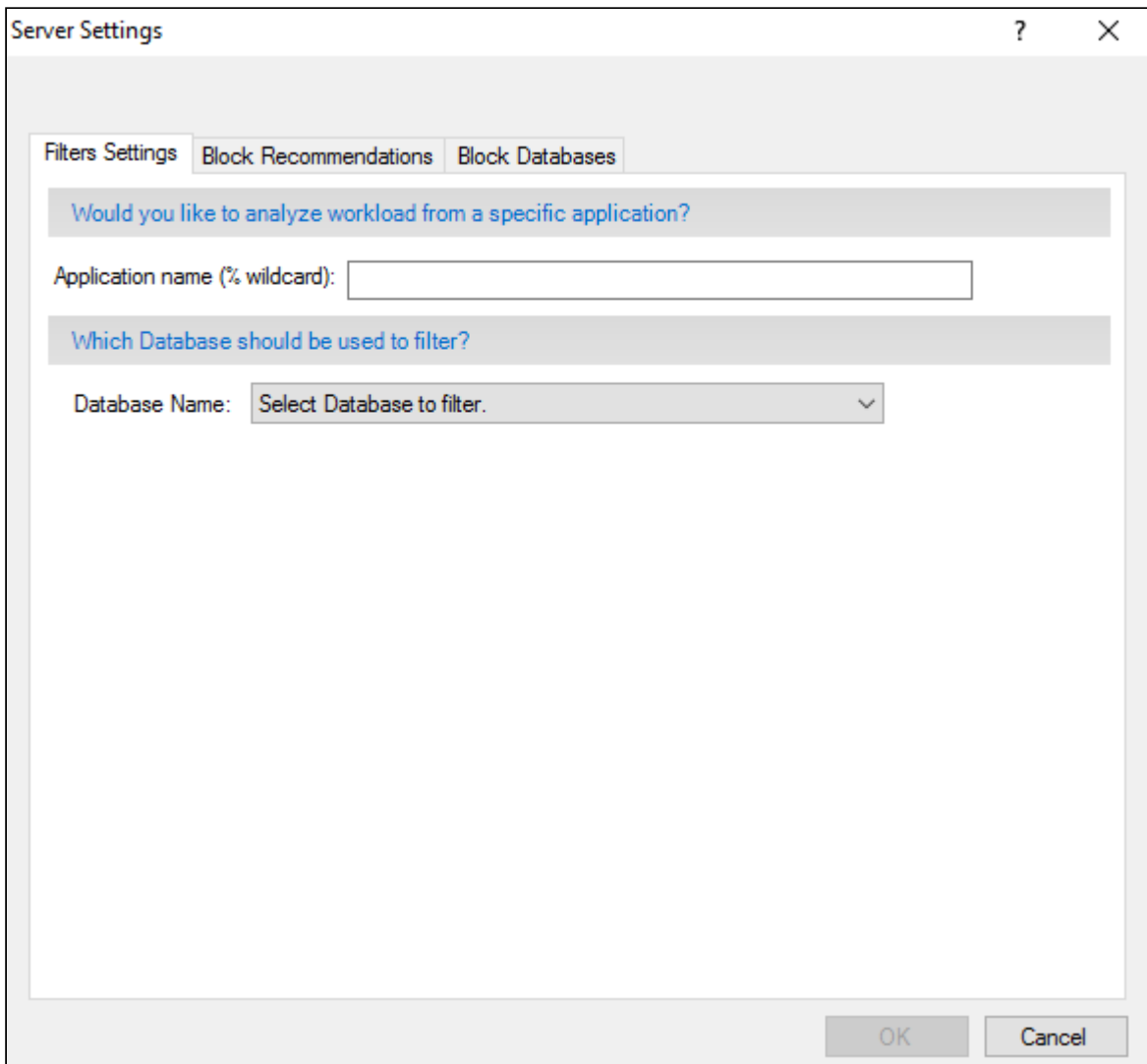
To cancel an analysis:

1. Right-click the appropriate monitored SQL Server instance and select **Properties > Analysis Configuration**.
2. On the Setup Analysis Categories and Options window, under the Scheduling section, clear the **Scheduling Enabled** check box.

Fine-tune your analysis

SQL Diagnostic Manager provides several options to fine-tune your prescriptive analysis results. You can apply filters that allow you to focus on a specific application or database. You can also reduce analysis "noise" by blocking recommendations and databases for changes that you do not plan to make in your environment.

Apply filters

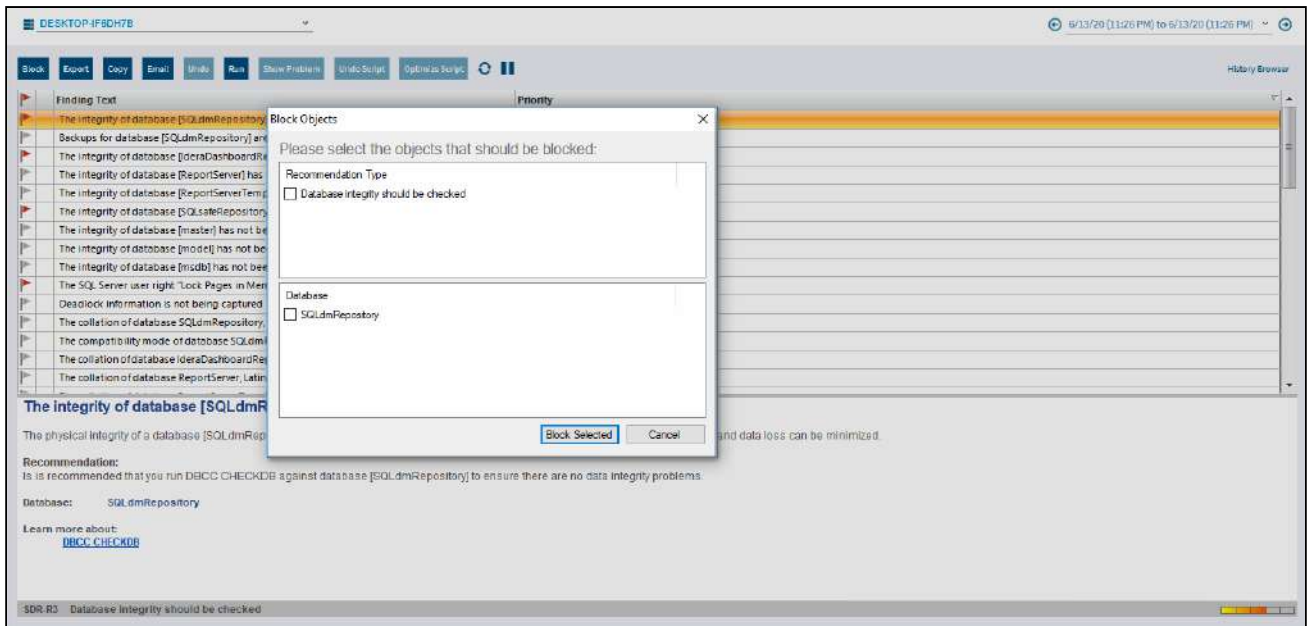


The screenshot shows the 'Server Settings' dialog box with the 'Filters Settings' tab selected. The dialog has three tabs: 'Filters Settings', 'Block Recommendations', and 'Block Databases'. The 'Filters Settings' tab contains two sections. The first section asks 'Would you like to analyze workload from a specific application?' and has a text input field labeled 'Application name (% wildcard):'. The second section asks 'Which Database should be used to filter?' and has a dropdown menu labeled 'Database Name:' with the text 'Select Database to filter.' and a downward arrow. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

To apply filters in your analysis results:

1. Right-click the appropriate SQL Server instance from the **Servers** sidebar and select **Properties**.
2. Select **Analysis Configuration** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.
3. Click **Advanced Settings** to access the Server Settings window.
4. Select the Filter Settings tab.
5. Enter the filter information in the provided fields, using (%) as a wildcard.
6. Click **OK**.

Block your recommendations

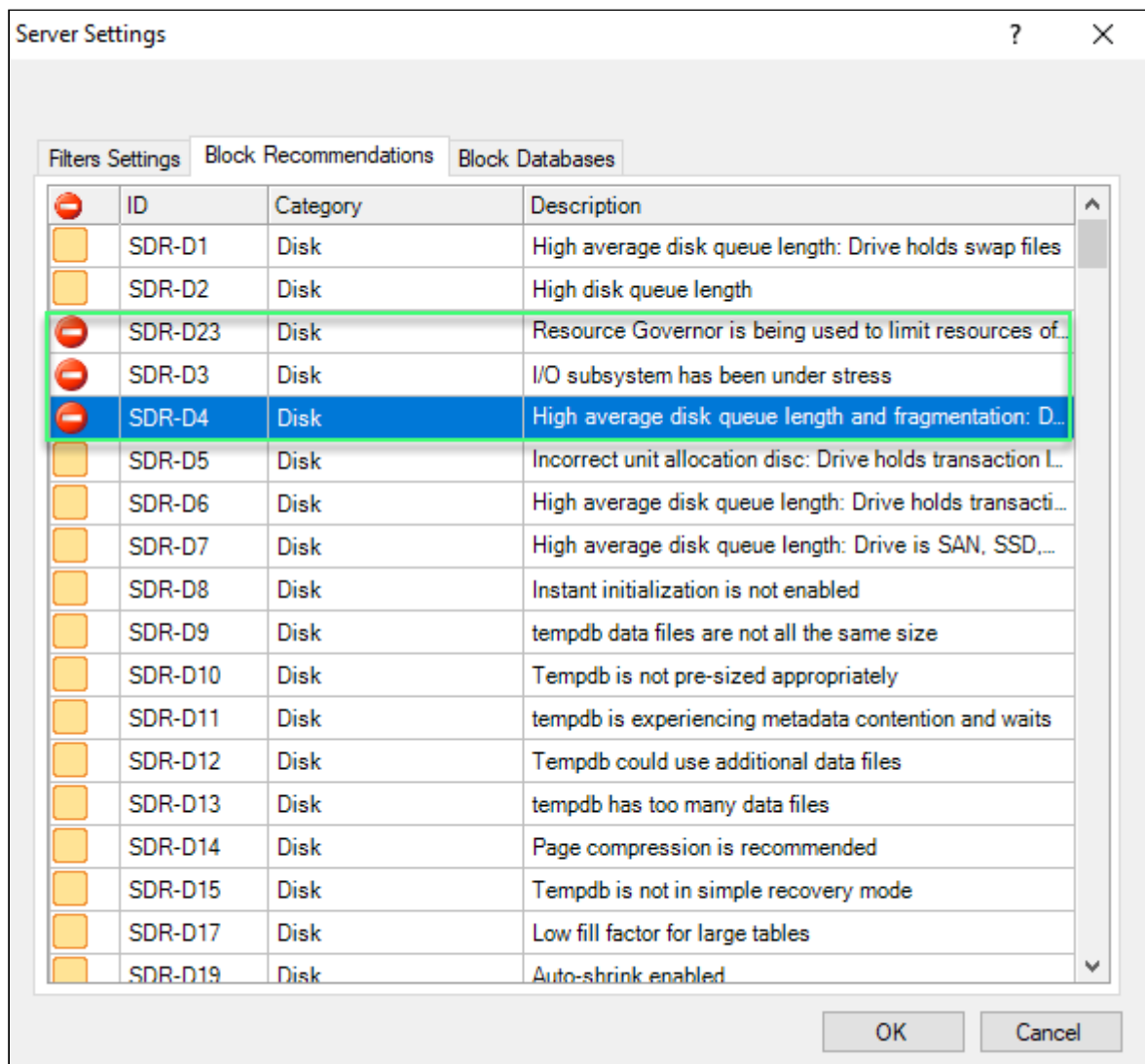


To block recommendations from appearing in your analysis:

1. Select one or more recommendations returned from the analysis.
2. Click **Block** in the **Actions** group of the Analyze tab and the Block Objects screen displays.
3. Select the recommendation check box or boxes under Recommendation type.
4. Click **Block Selected**.

To block recommendations from the Server Settings window:

1. Right-click the appropriate SQL Server instance from the **Servers** sidebar and select **Properties**.
2. Select **Analysis Configuration** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.
3. Click **Advanced Settings** to access the Server Settings window.
4. Select the Block Recommendations tab.
5. Select the **Block** check-box next to each of the recommendations you want to block from future analyses.
6. Click **OK**.



Blocking a recommendation removes it from all future analyses results for this registered instance.

Block databases

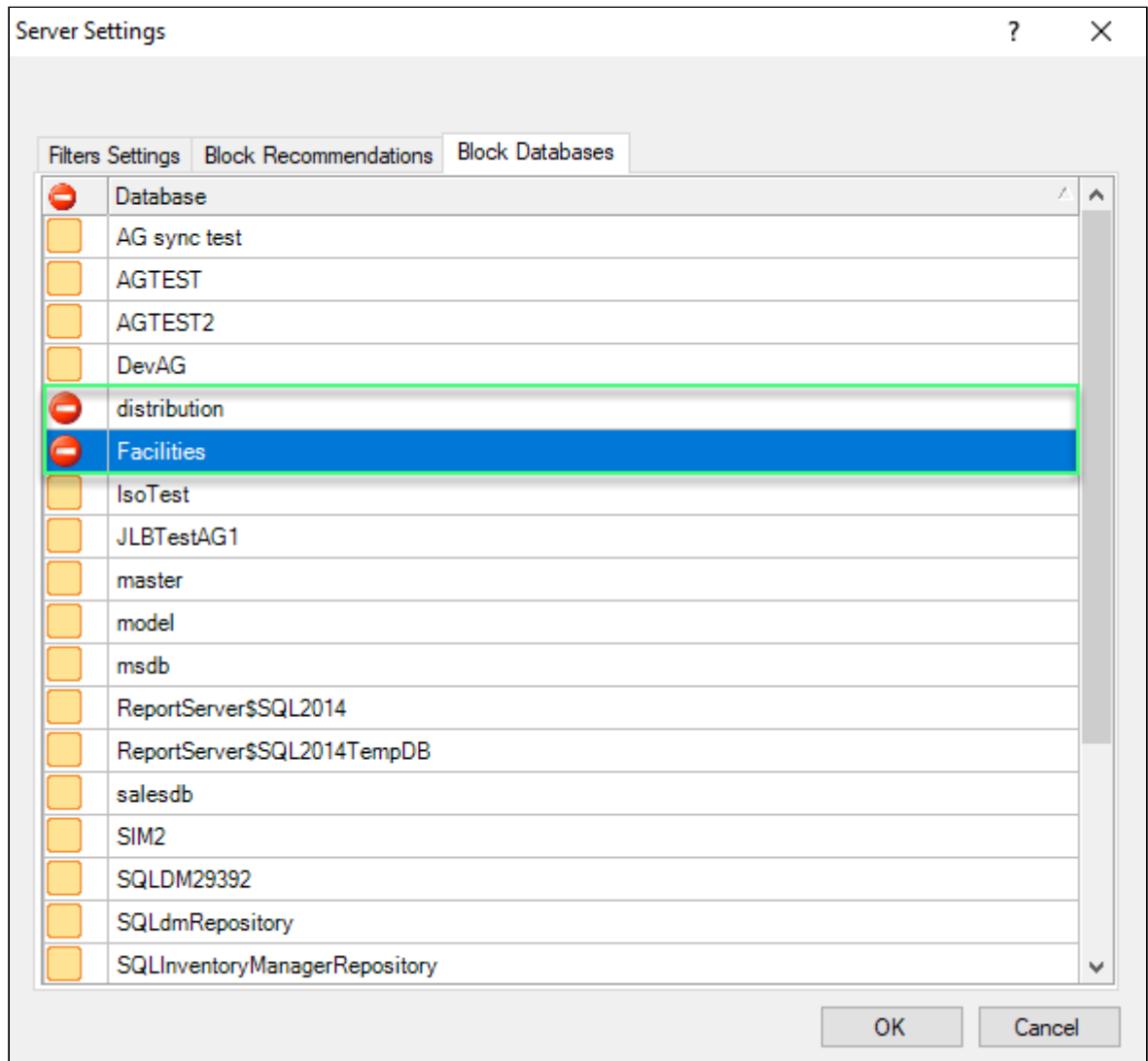
To block databases from appearing in your analysis:

1. Access the Block Objects screen by selecting one or more recommendations and clicking **Actions > Block** in the Analyze tab.
2. Select the database check box or boxes under Database.
3. Click **Block Selected**.

To block databases from the Server Settings window:

1. Right-click the appropriate SQL Server instance from the **Servers** sidebar and select **Properties**.
2. Select **Analysis Configuration** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.
3. Click **Advanced Settings** to access the Server Settings window.
4. Select the Block Databases tab.

5. Select the **Block** check box next to each of the databases you want to block from future analyses.
6. Click **OK**.



i Blocking a database removes it from all future analyses results for this registered instance.

Unblock your recommendations or databases

To unblock recommendations or databases to include them in your analysis:

1. Right-click the appropriate SQL Server instance from the **Servers** sidebar and select **Properties**.
2. Select **Analysis Configuration** when SQL Diagnostic Manager displays the Monitored SQL Server Properties window.
3. Click **Advanced Settings** to access the Server Settings window.
4. Select the Block Recommendations tab or the Block databases tab of the Server Settings window.
5. Scroll down the list of blocked recommendations or databases and find the recommendation or database you want to unblock.

6. Clear the **Block** box next to the appropriate recommendation or the **Block** box next to the appropriate database.
7. Click **OK**.

Identify a problem and optimize scripts

On the Analyze tab, in the Scripts group, you can identify a problem and optimize performance:

Show Problem

The **Show Problem** button launches the SQL Viewer window with the SQL text associated with the recommendation highlighted. Click **Copy** to copy the SQL text to the clipboard so you can view the text in a text editor or in SQL Server Management Services.

Optimize Script

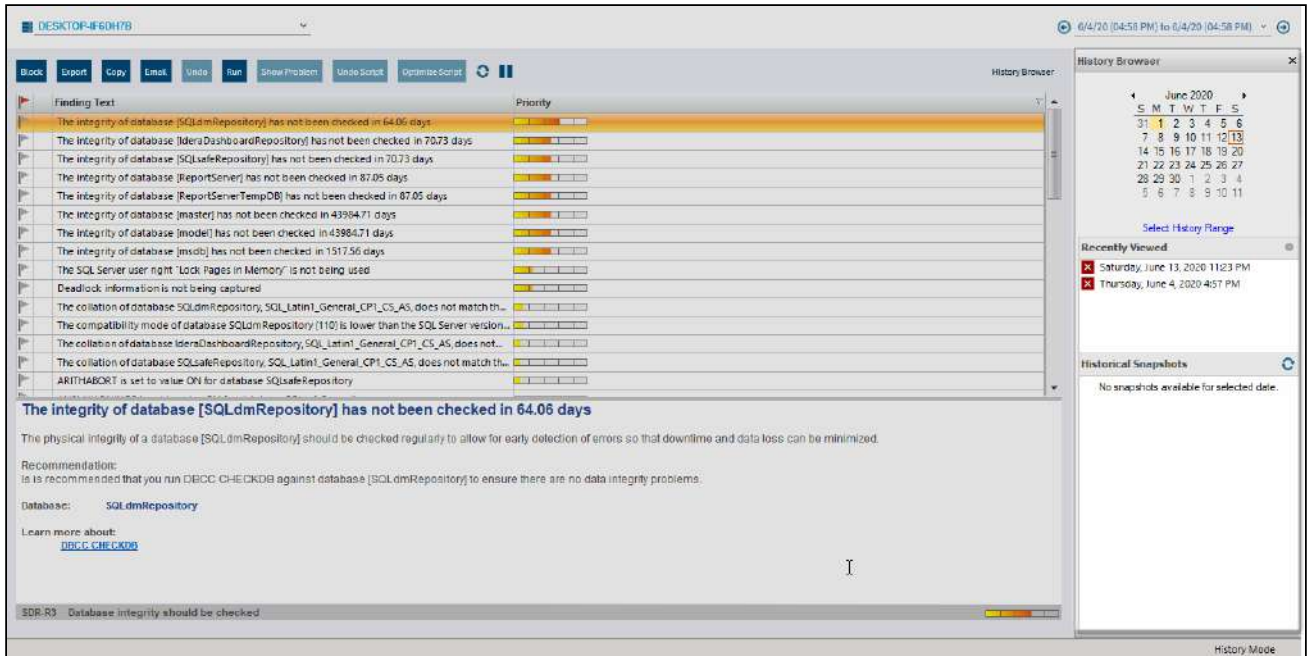
The **Optimize Script** window displays the script that is used to enact the recommended change. Review the script, and then either click the **Run** button to initiate the optimization or click **Cancel** to return to the recommendation you were viewing. You can also copy the script to Notepad or another text editor to examine the code and research other options further before making the change.

Undo Script

The **Undo Script** window displays the script that is used to enact the recommended change. Review the script, and then either click the **Run** button to initiate the undo or click **Cancel** to return to the recommendation you were viewing. The undo script can be copied to clipboard and used at a later time.

View your prescriptive analysis results

On the Analyze tab, in the History group, you can access the results of every analysis stored by SQL Diagnostic Manager for your monitored SQL Server instance.



Analyses are ordered by start time per the local computer running the SQL Diagnostic Manager console, and include:

- Time stamp when the analysis started.
- Total number of recommendations found.
- Highest priority recommendation associated with the analysis.

To select a specific analysis:

1. Open the [History Browser](#).
2. Select the date from the calendar.
3. Select the corresponding analysis from the Historical Snapshots list.

Additional actions

Previous Analysis

Browse to the previous analysis result of your SQL Server instance.

Next Analysis

Browse to the next analysis result of your SQL Server instance.

- ✓ To return to Real-Time mode, click **Run Analysis** on the Analyze tab.

Diagnose queries

Access query diagnosis from any SQL Diagnostic Manager view that contains a SQL query such as the Query Statement Mode, Signature Mode, or the alert views. Query diagnosis helps you save time by analyzing the queries you care about without having to wait for a full analysis to complete.

When SQL Diagnostic Manager diagnoses a query, you receive performance recommendations.

To a query diagnosis:

1. Right-click any selected SQL query and select **Show Query Text**.
2. In the SQL Text dialog, click **Diagnose** to start your query diagnosis.
3. In the Analyze tab a priority-ranked list of recommendations displays.

View the Timeline for this instance

The Timeline view provides a chronological chart of events that affect your monitored SQL Server instance. The view details the point in time for this monitored SQL Server instance when an alert occurred, including custom counter alerts.

SQL Diagnostic Manager displays as many alerts as can fit in the window and uses color codes based on the alert type as shown in the key below the menu. While a single Timeline point can represent more than one alert, the point itself appears in the color of the most critical alert for that time.

Use the filters to pinpoint the information you want to view. If the displayed timeframe is not exactly what you want to view, select another period from the **Filter** drop-down list. Notice that the measurement at the bottom of the window changes from hours to days if you switch from a daily option to one over a period of days. Select **Pick a Day** to select a specific day using the calendar.

You can also use the time frame slider at the bottom of the view to zoom in or out by dragging the start and end bars to the desired position. Click the left or right arrows to move your set range to another period in the timeline. SQL Diagnostic Manager retains the range size, but changes the results based on the dates within the range.

SQL Diagnostic Manager displays alerts on the Timeline in your local time even if the monitored SQL Server instance where the event occurred is located in a different time zone.

Access the Timeline view

You can open the Timeline view of the SQL Diagnostic Manager Overview tab by selecting the appropriate SQL Server instance, and clicking **Overview > Timeline**.

Access more alert detail from the Timeline

View the total number of alerts at a specific time

When you hover your cursor over a point in the timeline, SQL Diagnostic Manager displays the number of alerts for the time displayed in the timestamp.

Zoom in for more detail

Your alert configuration may cause the Timeline view to appear cluttered in the default settings. Occasionally, a timeline point appears without an attached alert flag to let you know about the associated alert. If you experience these issues, use the time frame slider at the bottom of the view to zoom in for a clearer picture of a set of alerts.

View alerts

While the Timeline provides your alerts in chronological format, you may want to view more detail about a single alert while using this view. To access the Alerts view from the Timeline, simply right-click the alert and select **View Alerts**. SQL Diagnostic Manager displays the view you receive when you click the Alerts pane for a monitored SQL Server instance.

DBaaS custom dashboard

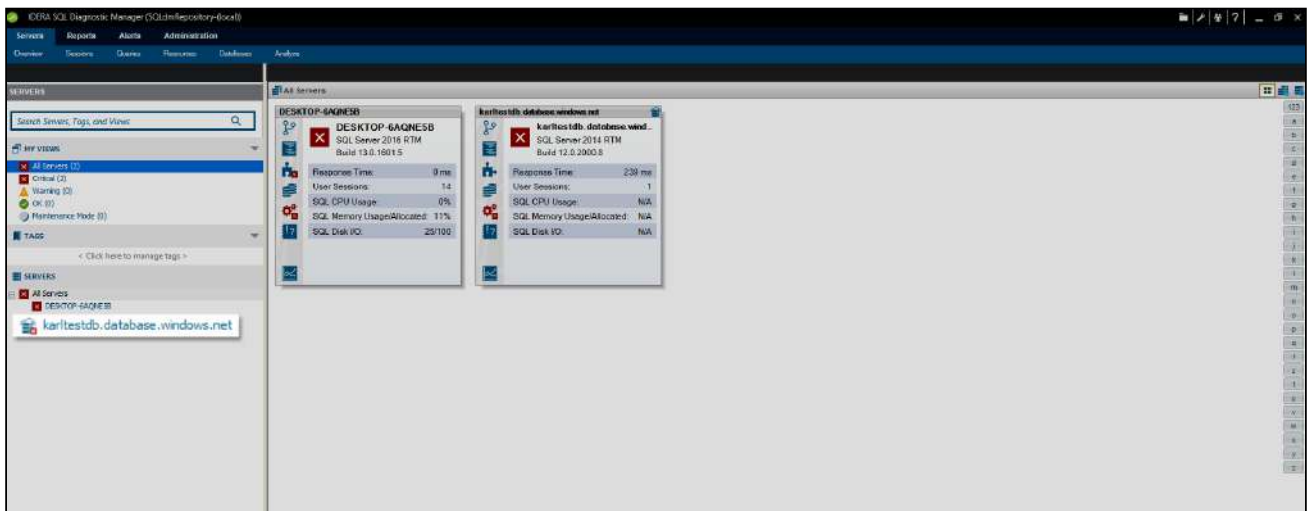
In the desktop console you can create a custom dashboard with metric panels that are supported in SQL Diagnostic Manager.

Azure SQL Monitoring

SQL Diagnostic Manager fully supports Azure SQL Databases and Azure Managed Instances. It includes the support for the following Azure SQL Deployment options:

- Azure SQL Single database
- Azure SQL Elastic Pools
- Azure SQL Managed Instance

Within the Azure coverage, SQLDM has a complete and custom navigation pane for Azure SQL monitoring. The Azure SQL and RDS servers are identified by a cloud icon  which helps users to recognize them easily.



The server navigation includes the following tabs when monitoring Azure SQL servers:

- [Overview](#)
- [Sessions](#)
- [Queries](#)
- [Resources](#)
- [Databases](#)
- [Analyze](#)

Azure Discovery

SQL Diagnostic Manager discovers all the Servers associated with your Azure Profile credentials, which allows you to easily add your Azure SQL servers.

1. Right-click All Servers located on the SERVERS side screen
2. Select **Manage Servers** adding a new server
3. Under the Manager Server window select the **Add** option
4. In the Welcome Add Servers Wizard, click **next**.
5. Select SQL Server Authentication and enter your credentials.
6. Select the **Enable Azure Discovery Settings** checkbox.

Add Servers Wizard ? X

Configure Authentication
Select the authentication mode and credentials SQL Diagnostic Manager should use when collecting diagnostic data from the monitored SQL Server instances.

i SQL Server instances are added in groups based on the authentication mode specified below. Please note that all diagnostic data is collected by the SQL Diagnostic Manager Collection Service. If you choose to connect using Windows Authentication, the account the SQL Diagnostic Manager Collection Service is running as will be used to connect to monitored SQL Servers.

Connect using:

Windows Authentication

SQL Server Authentication

Login name:

Password:

Enable Azure Discovery Settings

Advanced Encryption Options

Encrypt Connection (SSL)

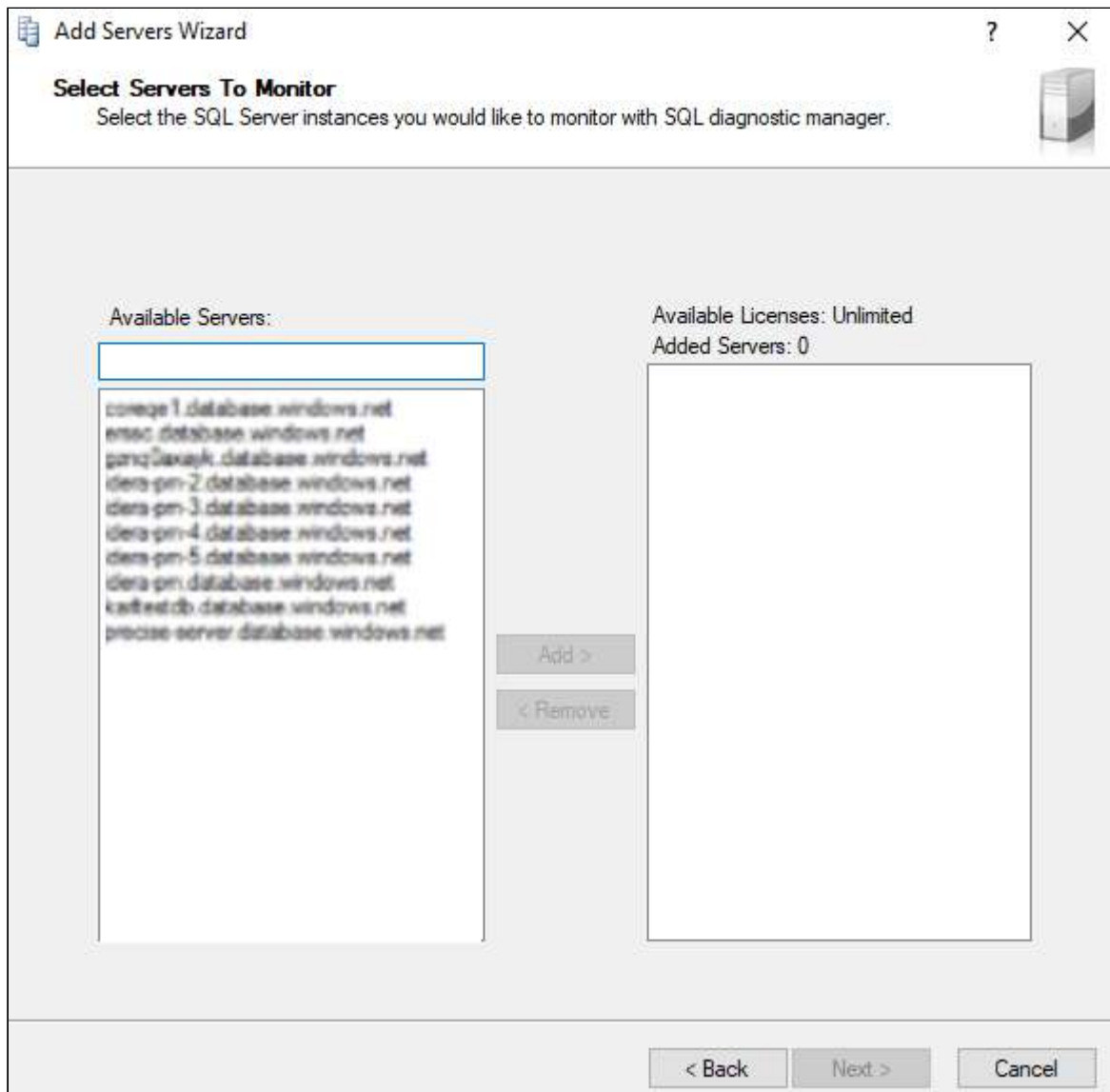
Trust Server Certificate (bypass certificate validation)

Azure Discovery Settings

Selected Azure Profile

< Back Next > Cancel

7. Click the **Azure Discovery Settings** button. When the Azure Application Configuration opens, enter your credentials, select the **Azure profile** from the drop-down list, and click **OK**.
8. The Azure profile selected displays in the Selected Azure Profile textbox, and click **Next**.
9. The result of your research should be all the servers related to your Azure Profile, check the image below. Select the servers you want to add, click **Add>** to move them to the Added Servers, and click **Next**.



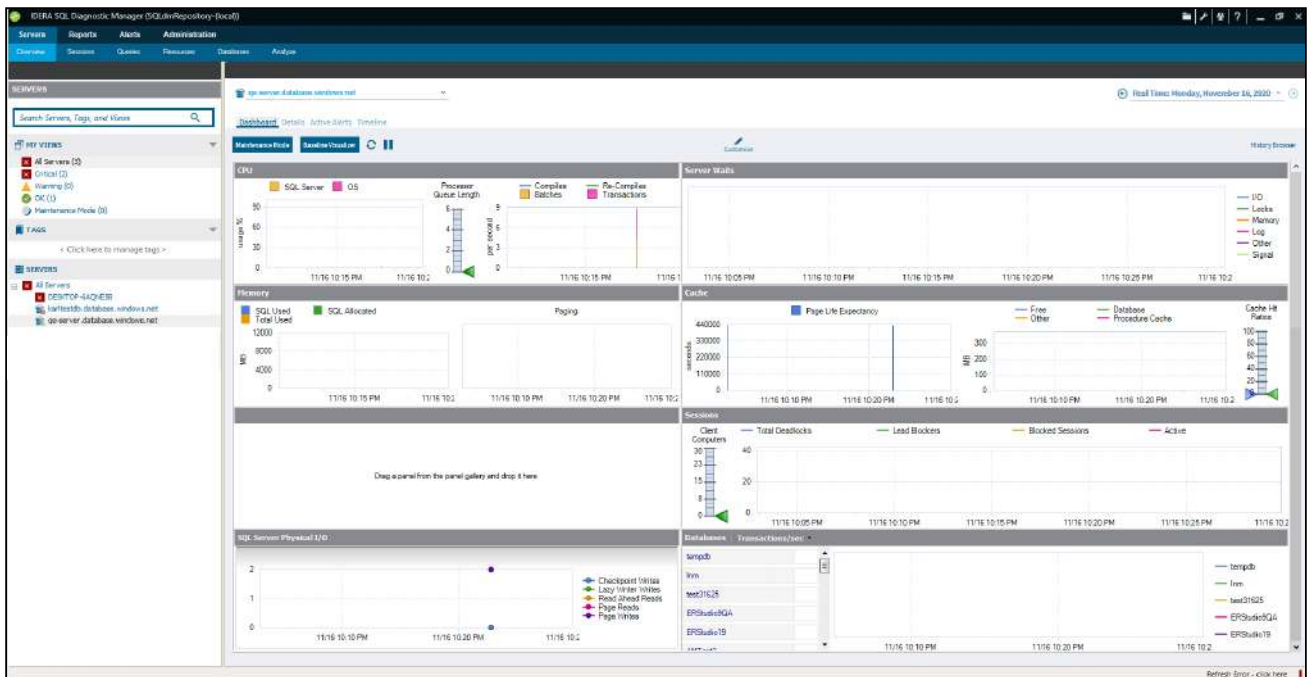
Azure Overview

Dashboard

The Dashboard view shows you an overview of the instance. The overview tab displays the following graphs, by default:

- CPU
- Server Waits
- Memory
- Cache
- Sessions
- SQL Server Physical I/O
- Databases

You can add or edit the layout of your Dashboard by clicking **Customize**. In the **Customize Dashboard** screen, click **View Panel Gallery**, and drag the panels from the right side to your Dashboard screen on the left.



Details

The Details screen shows the properties of the monitored Azure SQL server, including the Version, Edition, Host OS, Databases, and Log Size. On the right screen, you can review the metrics list, check any metric to add it in the chart located at the bottom of the screen.

Active Alerts

The Active Alerts view allows you to check a list of all the active alerts as of the last data collection. Click any alert in the list to view details at the bottom of the window.

Timeline

The Timeline view provides a chronological chart of events that affect your monitored SQL Server instance. The view details the point in time for this monitored SQL Server instance when an alert occurred, including custom counter alerts.

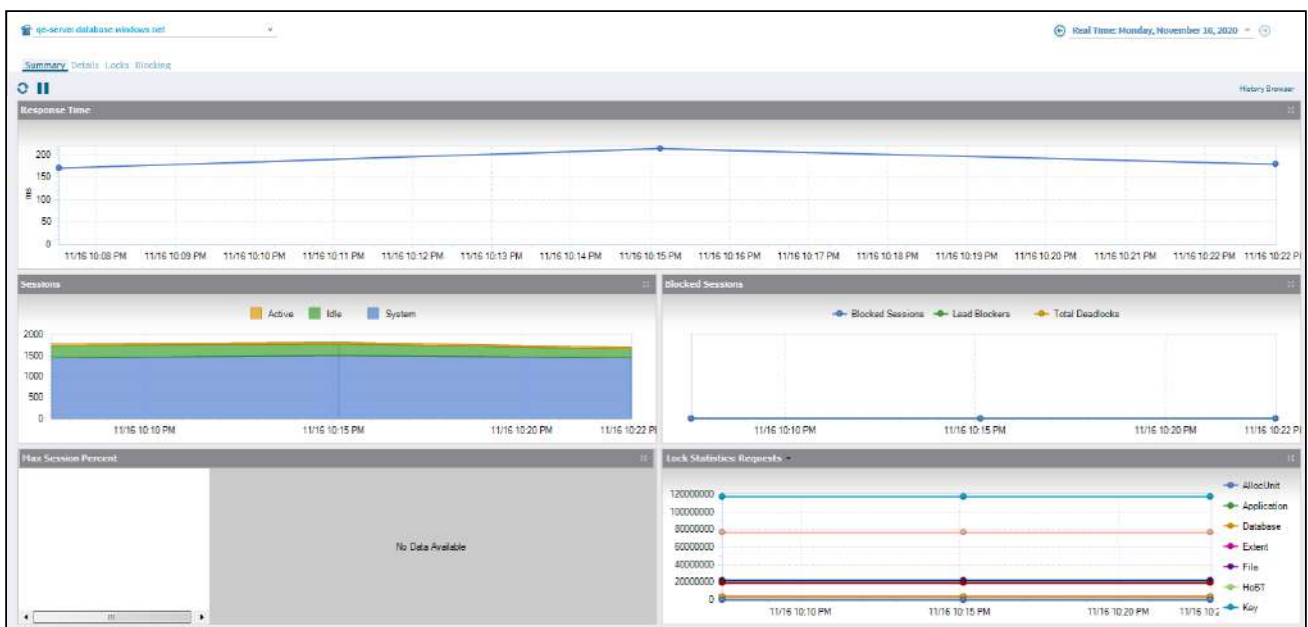
Azure Sessions

The Sessions tab includes the Summary, Details, Locks, and Blocking screens.

Summary

The Sessions Summary shows real-time charts for your monitored Azure SQL server, the following charts are displayed in this view:

- Response Time
- Sessions
- Blocked Sessions
- Max Session Percent
- Lock Statistic



Details

The Sessions Details view provides an in-depth analysis of sessions running on your monitored Azure SQL server.

Locks

On the Locked Sessions view, you can see which locks are causing data availability problems and unacceptable responses. At the bottom of the screen, the [Lock Statistics](#) chart has the same function as a non-cloud monitored server.

Blocking

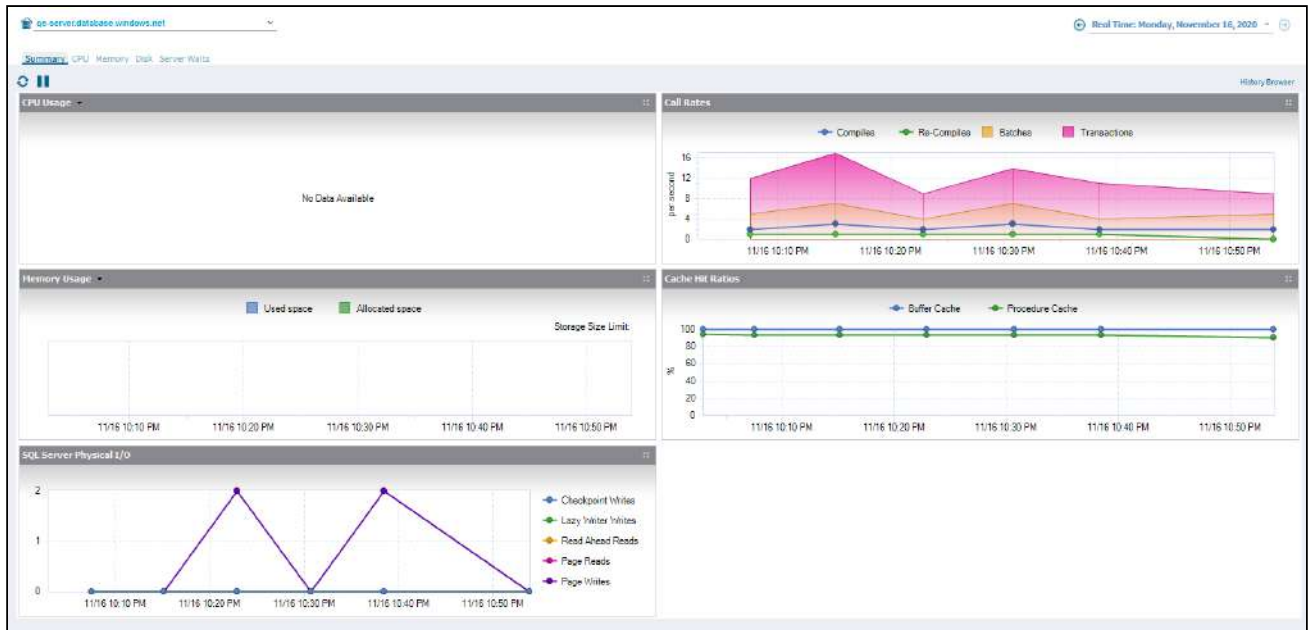
The Blocking view provides a blocking tree, along with Block Reports, Deadlock Reports information. At the bottom of the screen, the Block Sessions chart represents the Blocked Sessions, Lead Blockers and Total Deadlocks information.

Azure Resources

The Resources tab allows you to monitor the system resources on the server hosting the Azure SQL instance.

Summary

The Resources Summary tab displays real-time charts allowing you to quickly identify key diagnostic statistics for monitoring the resources for the selected Azure SQL instance.



CPU

The Call Rates chart displays the rate of batch statements, the compiles into procedure cache, and the recompile of statements stored in the procedure cache.

In addition, the Max Worker Percent chart shows the percentage of worker threads used on the database, relative to the maximum available for the service tier.

Memory

The following charts are available on the Memory tab:

- SQL Memory Usage - Displays the SQL Memory Usage by what is used by SQL Server compared to what is allocated and the total used.
- Paging - Displays the memory pages per second.
- Memory Areas - Displays the memory usage of the Procedure Cache, Connections, Locks, Database, Optimizer, Sort, Hash, and Index.
- Cache Hit Ratios - Displays the hit ratios of the Buffer and Procedure cache.
- Buffer Cache - Displays the buffer cache used by Free and Active pages.
- Page Life Expectancy (sec) - Provides the time in seconds that a page remains in the cache or buffer pool.
- Database Average Memory Usage Percent - Displays the percentage of memory used for the database, relative to the maximum available for the service tier.

Disk

The Disk tab shows the key statistics on the way your disks are used on your Azure SQL server.

Server Waits

The Server Waits view allows you to see all of the waits affecting Azure SQL server performance.

Azure Databases

SQL Diagnostic Manager provides a graphical analysis of the used, unused, and index disk space of the database. You can also view the size and structure of database tables, this screen provides a list of data size, text or image size, index page size, and the number of table rows, making reorganization and index creation decisions easy.

Summary

The Database Summary view allows you to view the status of all the databases on the selected Azure SQL server.

The following graphs are displayed at the bottom of the screen:

- Capacity Usage - Select how to view the data or the log is used on your database. The drop-down list allows you to choose to view the data or log in megabytes or percentages.
- Recent Trends - Select how to view how your database or group of databases are used currently by selecting either active sessions, transactions per second, data size, log size, or log flushes from the drop-down list.

Configuration

The Database Configuration view allows you to review the configuration settings of the databases on the selected Azure SQL instance. This view contains a single grid that includes the configuration settings for all databases on the selected SQL Server instance.

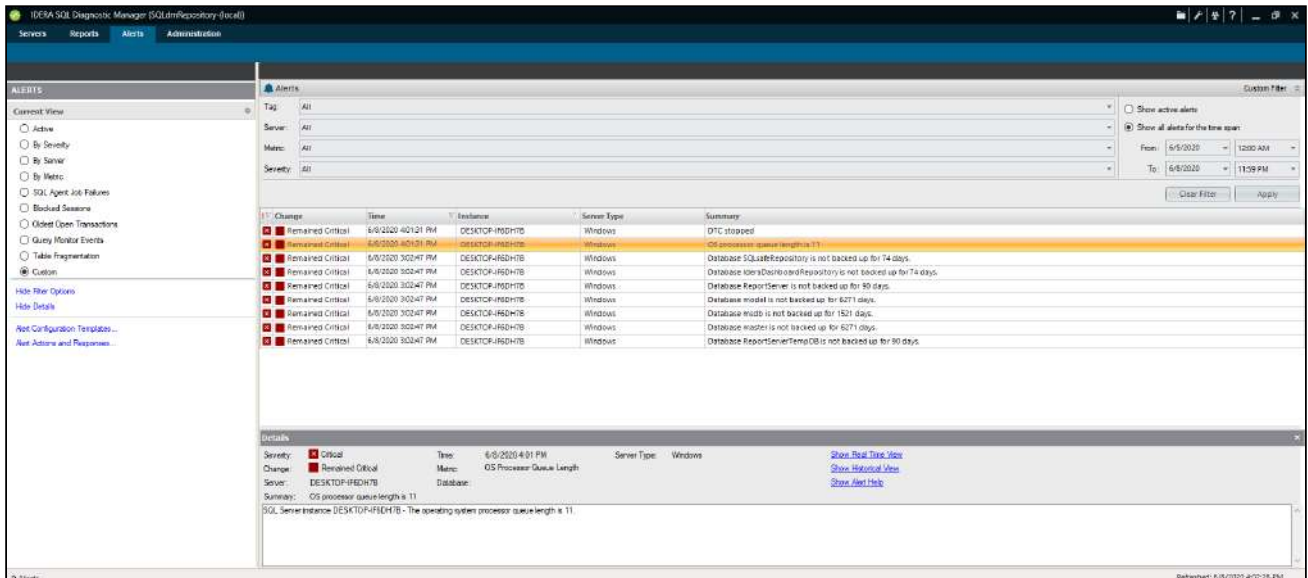
Azure Analysis

The Analyze tab runs prescriptive analyses on Azure SQL servers to identify and resolve SQL Server performance problems. The analysis engine scans your SQL Server configuration for potential problems and the health of your databases, resulting in a useful set of recommendations for improving performance for your monitored Azure SQL servers.

SQL Diagnostic Manager includes compatible recommendations for your monitored Azure SQL instances. Follow the same steps of a windows server analysis for an Azure SQL server, [Run a prescriptive analysis on your SQL Server](#).

Organize alerts

Access the Alerts view from the SQLDM desktop console by clicking the **Alerts** tab.



Organize your alerts by Active Alerts

The Alerts tab allows you to view all the alerts for all of the monitored servers. Under the Alerts screen you can review all the active alerts; additionally, you can filter them by selecting an specific time frame.

Tag, Server, Metric, and Severity are the main custom filters you can configure under the Alerts screen, use **Apply** to retrieve the alerts under the configuration you choose or use **Clear Filter** to remove all the configured filter options.

- ✓ The **Change** column is a useful way to see the agent history of a metric. This column tracks every status change occurring between OK, informational, warning, and critical states.

Organize your alerts by the Current View

You can organize your alerts by selecting one of the following views from the Current View section of the Navigation Pane:

Active

Organizes all your alerts that are currently active.

By Severity

Organizes all your alerts by their severity.

By Server

Organizes all your alerts by monitored SQL Server instance.

By Metric

Organizes all your alerts by each of the metrics affected.

Agent Job Failures

Lists all the alerts caused by Agent Job failures.

Blocked Processes

Lists all the alerts caused by blocked processes.

Oldest Open Transactions

Lists the oldest open transactions that have alerts associated with them.

Query Monitor

Lists all the alerts associated with the Query Monitor.

Table Reorganization

Lists all the alerts associated with table reorganization.

Custom

Allows you to configure the view to show what is most important to you.

Organize your alerts using the Filter Options

SQL Diagnostic Manager allows you to organize your alerts using the available filter options. Click the **Show Filter Options/Hide Filter Options** toggle in the Navigation Pane to open the Alerts filter. Filter options such as the specific view, SQL Server instance, metric, severity, and time range associated with the alert help pinpoint to data you want to view.

 Additionally, you can access the [Alert Configuration Templates](#) and the [Alert Actions and Responses](#) configuration options.

Alert on SQL Server Metrics

SQL Diagnostic Manager allows you to configure alerts to inform and warn you about approaching issues with your SQL Server instances. You can view these alerts using the SQL Diagnostic Manager Console, the IDERA Newsfeed, or SQLDM Mobile.

When an alert threshold is reached, SQL Diagnostic Manager can:

- Send an email notification
- Pop up an alert message in your Windows taskbar
- Write an event to the Windows Event log
- Generate an event on the Timeline
- Send the alert message to the IDERA Newsfeed Action Provider, which publishes the alert message as a status update on the server wall in the IDERA Newsfeed interface and on the Active Alerts view in SQLdm Mobile.
- Create a new To Do item

To Do feature

Beginning with SQL Diagnostic Manager 7.0, the To Do feature is available only to users who had To Do activated in an earlier 6.x version. This version of SQL Diagnostic Manager does not support this feature for new users.

In response to a set threshold entering the error level, SQL Diagnostic Manager can perform one or more of the following actions:

- Send an email notification
- Write an event to the Windows Event log
- Generate an event on the Timeline
- Send the alert message to the IDERA Newsfeed Action Provider, which publishes the alert message as a status update on the server wall in the IDERA Newsfeed interface and on the Active Alerts view in SQLdm Mobile.
- Create a new To Do item

Once you correct the situation triggering the alert, SQL Diagnostic Manager alerts you again if the situation recurs.

- ✓ On each SQL Diagnostic Manager view, columns containing associated alerts are highlighted with their current status color. You can right-click many of these columns and select **Configure Alert** to change the associated thresholds.

Database- or disk-level alerts


SQL Diagnostic Manager includes some alerts that allow you to set them at a database or disk level, meaning that you can set different alert thresholds for each database or disk within a monitored SQL Server instance.

Alert templates

SQL Diagnostic Manager allows you to configure generic alert settings as a template that you can apply to servers and groups of servers in your organization. Click **Tools > Alert Configuration Templates** and complete the required fields to [configure an alert template](#).

Informational alerts

Informational alerts allow you to set a threshold that, when generated, triggers a status that does not affect the overall status of the server within SQL Diagnostic Manager. You can use informational alerts to notify an administrator of the state of a particular metric for a server or trigger secondary processes that could take action to prevent issue escalation.

 Note that for certain metrics, using the informational alert means that you no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.

IDERA Newsfeed

The IDERA Newsfeed is a revolutionary new way for DBAs and managers to collaborate, share knowledge, and keep close tabs on your most critical SQL Server issues. With the IDERA Newsfeed, you can quickly share information to stay informed, be more productive and shorten the time to problem resolution.

For more information, see the [IDERA Newsfeed Help](#).

SQLDM Mobile

SQLDM Mobile includes the IDERA Newsfeed as well as several dashboard views that define your server health and expose critical alerts. You can access the SQLDM Mobile Web application from most smart phones and mobile devices.

Configure alerts

The Alert Configuration window allows you to set up alerts for specific SQL Server instances. To open the Alert Configuration window with a SQL Server instance selected, right-click the instance, and then select **Configure Alerts**. You can set the acceptable thresholds using the following alert types:

Alert Type	Color	Meaning
OK	Green	Acceptable threshold where SQL Diagnostic Manager does not generate an alert.
Informational	Blue	Informational threshold where SQL Diagnostic Manager generates an informational alert.
Warning	Yellow	Warning threshold where SQL Diagnostic Manager generates a warning alert.
Critical	Red	Critical threshold where SQL Diagnostic Manager generates a critical alert.

In addition to setting the thresholds for specific alerts, you can customize your alerts by adding comments to each alert. Comments allow you to include instructions to users when a level reaches a particular threshold or to provide additional information about the alert. You can also customize the levels at which SQL Diagnostic Manager provides alert notifications by changing the informational, warning, and critical values either on the Configuration tab of the Alert Configuration window or, for database- and disk-level alerts, on the Database Threshold Configuration window available by selecting the database or disk, and clicking **Edit**.


You can also base your alerts on the past performance of the metrics collected by SQL Diagnostic Manager. This is a powerful and effective way to make sure that the alerts you receive are outside of your typical metric ranges.

- ✓ For each alert metric, you can enter the specific criteria by double-clicking the values on the Configuration tab.

To configure an alert:

1. Right-click a SQL Server instance in the Servers sidebar.
2. Select **Configure Alerts**.
3. Select the metric you want to edit from the list in the **Alert Configuration** window.
4. **If the metric features a per-database or per-disk alert**, on the Configuration tab of the Alert Configuration window, click **Add**. Use the drop-down list to select the database or disk to which you want to apply these settings. **If the metric applies at the instance level**, continue with the next step.
5. Check the boxes next to **Informational**, **Warning**, and **Critical** to include alerts for these states.
6. Change the alert thresholds by moving the arrows to the appropriate levels or by double-clicking the value and typing a new threshold level.
7. **If the metric features a per-database or per-disk alert**, click **Advanced** to apply any advanced settings, such as [alert suppression](#) or [autogrow settings](#). Click **OK**. **If the metric applies at the instance level**, continue with the next step.
8. Select the **Comments** tab and enter information you want displayed in the alert message for this metric.

9. Click **Apply**.
10. **If you want to replicate these edits to other SQL Server instance SQL Diagnostic Manager is monitoring, tags, or templates**, click **Yes**.
11. Click **OK** to accept your changes.

 Note that when you modify an alertable metric in the Alert Configuration window, the **Create Template** button is disabled. Once you apply the modification, this command button is enabled again.

Alert refresh

A full refresh of alerts excluding table fragmentation occurs in the following circumstances even if longer collection intervals are defined:

- when the collection service starts
- when the collection service receives a new workload, such as redirecting the Management Service to a new Repository
- when you add a SQL Server for collection
- when a SQL Server resumes from Maintenance Mode
- when a user selects Refresh Alerts

This refresh does not cause collection of non-alertable data, nor cause collection of anything that is disabled.

Table Fragmentation alert thresholds


SQL Diagnostic Manager updates Table Fragmentation alerts with new threshold information only when new data is generated during the table statistics collection interval. It is preferred that you set your table statistics collection interval to occur when your server is not performing any production tasks.

Because this data collection occurs at most once per day and you can configure it to run as infrequently as once per week, you may experience a significant delay before new alerts are generated with your new alert threshold. You can [configure your wait monitoring using the Monitored SQL Server Properties window](#).

Apply different alert thresholds to multiple databases or disks

Some of the available alerts allow you to apply them independently to multiple databases or disks on your monitored SQL Server instance. When you click the alert to modify the thresholds, notice that a list of your databases or disks appears with the associated thresholds as they are currently set. Click **Edit** to make your changes. The [Database Threshold Configuration dialog](#) lets you specify the database, and then select the thresholds for that database only. The [Disk Threshold Configuration dialog](#) provides the same functionality, only disks replace the databases.


Alert templates

SQL Diagnostic Manager allows you to configure generic alert settings as a template that you can apply to servers and groups of servers in your organization. Click  > **Alert Configuration Templates** and complete the required fields to [configure an alert template](#).

Informational alerts

Informational alerts allow you to set a threshold that when generated triggers a status that does not affect the overall status of the server within SQL Diagnostic Manager. You can use informational alerts to notify an

administrator of the state of a particular metric for a server or trigger secondary processes that could take action to prevent issue escalation.

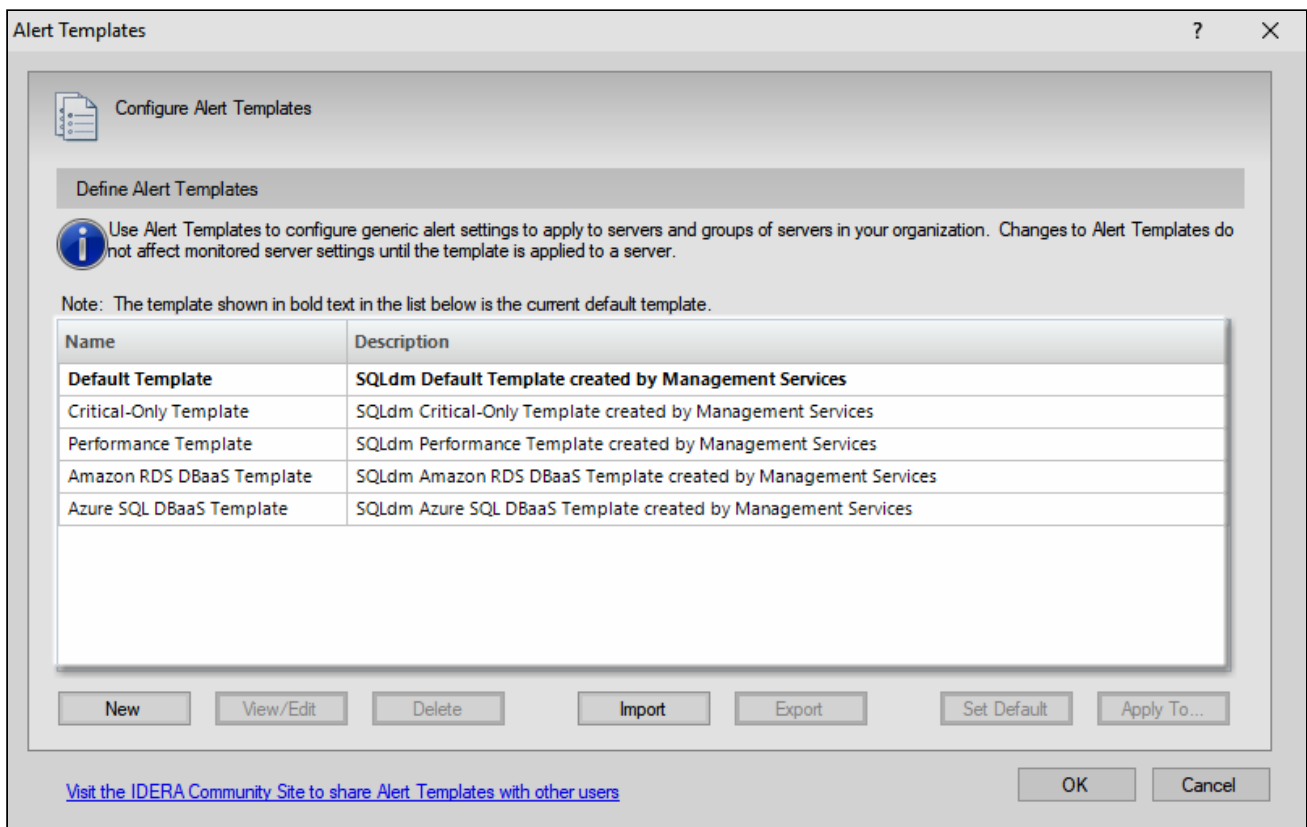
 Note that for certain metrics, using the informational alert means that you no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.

Configure alert templates


The Alert Templates window allows you to configure generic alert settings that you can apply to servers and groups of servers in your organization. An event that causes a metric to no longer meet a condition or to exceed a threshold signals SQL Diagnostic Manager to raise an alert. The Alert Templates save you time and help you to avoid mistakes by simplifying alert configuration across your environment.

SQL Diagnostic Manager has the following five Alert Templates that can be used to configure generic alert settings to apply to your monitored servers:

- Default Template
- Critical-Only Template
- Performance Template
- Amazon RDS DBaaS Template
- Azure SQL DBaaS Template




Applying an alert template to a monitored server overwrites all existing alert configurations. You can make any custom changes to the alert configuration once the template is applied. If you make a change to a template that is applied to a server, you must also reapply the template to that server using the **Apply To** button.

 Servers are not permanently attached to an applied alert template. SQL Diagnostic Manager does not propagate to the monitored server any changes made to the template unless explicitly applied.

A template allows you to apply the same alert configurations to more than one SQL Server instance. It helps you avoid simple mistakes such as transposing characters or leaving out metric details that could occur when you manually configure each server. Alert templates also help you roll out changes by making the updates in a single

location and then rolling out the changes across your environment. If you are more interested in ad hoc alert configuration, remember that you can use a template to return a server to default settings if you make any changes that you later want to revert.


Access the Alert Templates window


You can open the Alert Templates window from the SQL Diagnostic Manager Console menu bar by selecting  > **Alert Configuration Templates**.

Create a new alert template

Create a new alert template or edit the existing Default Template to suit your needs when alerting on your monitored SQL Server instances.

To create a new alert template:

1. Select  and click **Alert Configuration Templates**.
2. Click **New**.
3. Type a unique name and description.
4. **If you want to create a new template based on an existing template**, select the name of the existing template from the **Copy from** list.
5. **If you want to use this new template as your default template**, check **Default Template**.
6. When you click **OK**, the **"Would you like to apply this template to existing SQL Server instances?"** message prompts. Click **Yes** to confirm or click **No** if you do not want to apply this template.
 - a. If you click **Yes**, the Select Servers or Tags opens. Select Tags or Select Servers to which the alert template should be applied.

 You can edit the configuration of your new alert template. In the Add Alert Template window, click **Edit configuration** to change thresholds and other metric parameters, for more information review [Configure your alert settings](#).

Change an existing template

SQL Diagnostic Manager allows you to make changes to an existing alert template. Select the template from the list, and then click **View/Edit**. If necessary, you can delete any existing template except for the Default Template. To delete a template, select the template you want to delete, and then click **Delete**.

Import an existing template

SQL Diagnostic Manager allows you to [import](#) an existing alert template from the [IDERA community site](#). Click **Import** and select an alert template.

Export an existing template

SQL Diagnostic Manager allows you to [export](#) an existing alert template. Select the template from the list, and then click **Export**. You can use the default location or specify a different path to save the export file.

Apply template changes to a tag or server

Once you create a new template or make changes to an existing template, you should apply the updates to your tags or servers. Select the template from the list, click **Apply To**, select a tag or servers to which you want to apply the alert template, and then click **OK**.

Note that if the template you are making changes to is already associated with a tag or server, these changes are not automatically applied when you change the template. You must use the **Apply To** button and select the appropriate tag or servers.

Set a template to use as the default

You can set an existing template as the default for SQL Diagnostic Manager to apply to all new servers added without specifying an alert template. To designate a template as the default, click the template in the list, click **Set Default**, and then click **OK**. If there is only one template in your list, SQL Diagnostic Manager uses this template as the default.

Apply alert templates to individual databases or disks within an instance

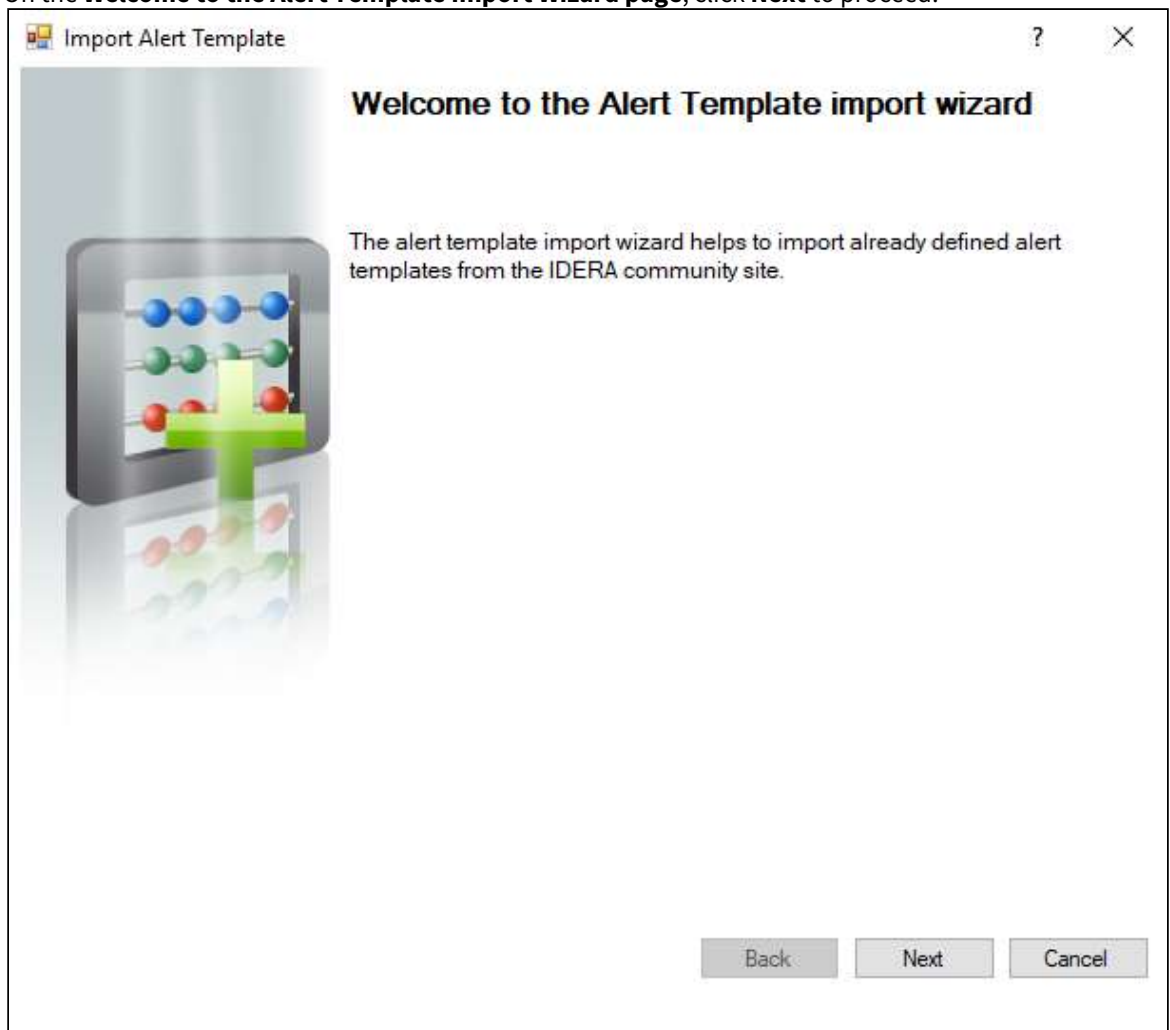
SQL Diagnostic Manager includes the ability to apply some alert thresholds on a per-database or per-disk basis within an instance. When you create an alert template and include one of these alerts, you cannot select which database or disk to apply the alert to because that is unknown in a template. For that reason, these alerts apply to all databases and disks within the instance to which it is applied.

Import an existing template

The Alert Templates window allows you to configure generic alert settings that you can apply to servers and groups of servers in your organization. Alert templates save you time and help you to avoid mistakes by simplifying alert configuration across your environment. SQL Diagnostic Manager allows you to import an existing alert template from the [IDERA community site](#) to suit your needs when alerting on your monitored SQL Server instances.

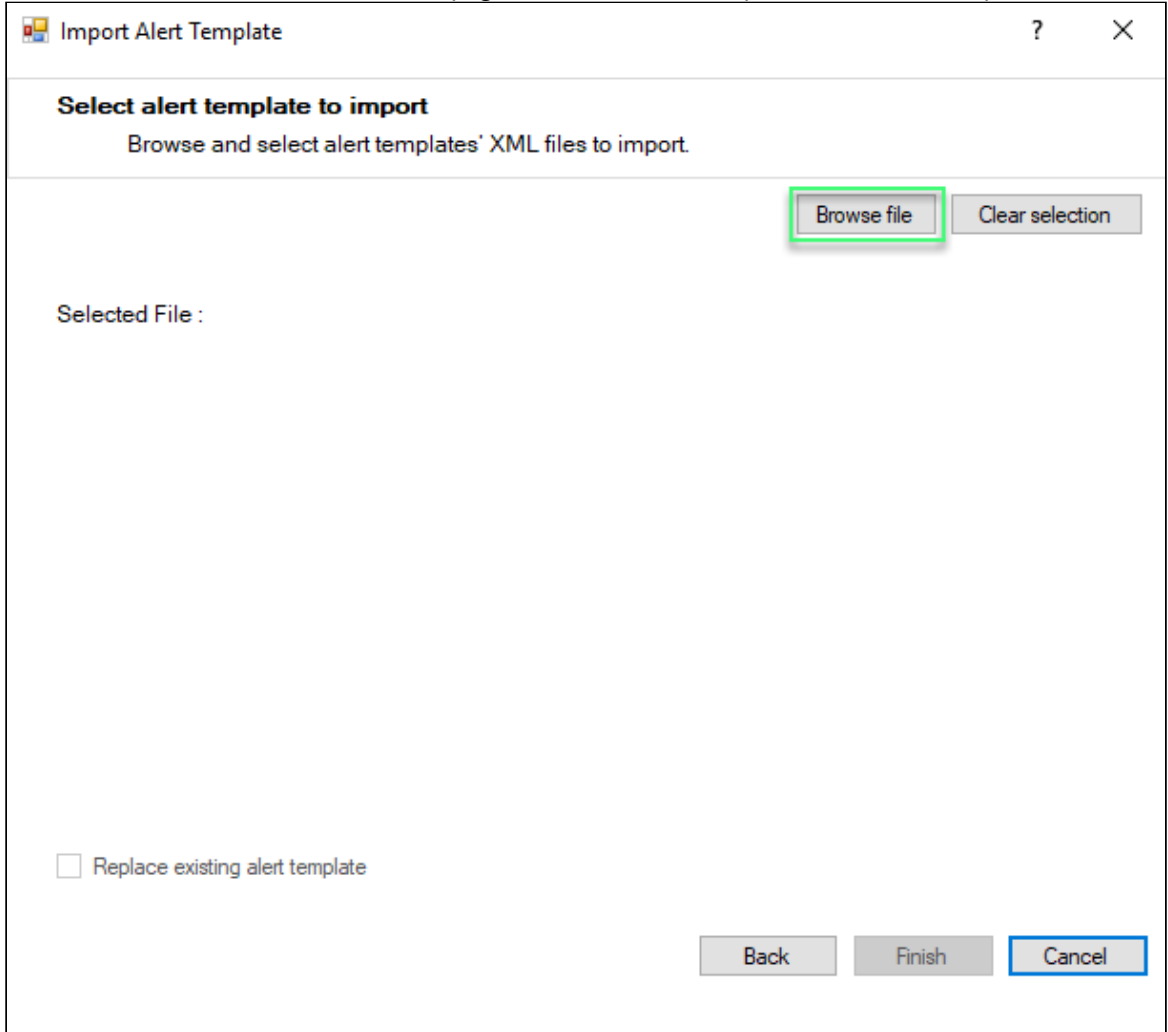
To import an alert template:

1. Click  select **Alert Configuration Templates**, and click **Import**.
2. On the **Welcome to the Alert Template Import Wizard page**, click **Next** to proceed.



- ✓ You can also access the Alert Template import wizard by clicking **Administration > Import/Export** from the navigation pane and selecting Alert Template in the import column.

- On the **Select alert template to import** page, browse for alert templates' XML files to import.



Import Alert Template

Select alert template to import
Browse and select alert templates' XML files to import.

Browse file **Clear selection**

Selected File :

Replace existing alert template

Back **Finish** **Cancel**

- If the imported alert template is valid, click **Finish** to complete the import process and exit the wizard.
- If you imported an alert template already in use (same name). You can click **Clear selection** and browse for a new alert template file. Alternatively, if you wish to replace your current alert template with the imported one, select the **Replace existing alert template** check box.
- If you imported an invalid alert template. You can click **Clear selection** and browse for a new alert template file.
- Click **Finish** to complete import process or **Back** to modify any settings.


✔ **Export an alert template**

For instructions on exporting an alert template, see [Export an existing template](#).

Export an existing template

The Alert Templates window allows you to configure generic alert settings that you can apply to servers and groups of servers in your organization. Alert templates save you time and help you to avoid mistakes by simplifying alert configuration across your environment. SQL Diagnostic Manager allows you to export and share an existing alert template in the [IDERA community site](#) to suit your needs when alerting on your monitored SQL Server instances.

To export an alert template:

1. Select  > **Alert Configuration Templates**.
2. Select the template from the list and click **Export**.
3. In the **Save As** dialog box, indicate in which folder and under what name to save the alert template XML file.
4. In the "Selected Alert Template Exported Successfully" notification, click **OK**.

To export all your alert templates at once:

1. Select **Administration > Import/Export** and click **Alert Template** in the export column.
2. In the **Browse For Folder** dialog box, indicate in which folder you want to save the alert templates XML files.
3. Click **OK** in the "Alert templates exported successfully" notification.




Import an alert template

For instructions on importing an alert template, see [Import an existing template](#).

Configure alert template settings

The Alert Template Configuration window allows you to set up the alerts templates for multiple or all SQL Server instances.


 Be sure to review a list of the [default alert IDs](#), their descriptions, and associated events before configuring your alerts.

You can set the acceptable thresholds using the following alert types:

Alert Type	Color	Meaning
OK	Green	Acceptable threshold where SQL Diagnostic Manager does not generate an alert.
Informational	Blue	Informational threshold where SQL Diagnostic Manager generates an informational alert.
Warning	Yellow w	Warning threshold where SQL Diagnostic Manager generates a warning alert.
Critical	Red	Critical threshold where SQL Diagnostic Manager generates a critical alert.

In addition to setting the thresholds for specific alerts, you can customize your alerts by adding comments to each alert, and customize the alert rank based on the users environment and preferences. Comments allow you to include instructions to users when a level reaches a particular threshold or to provide additional information about the alert. You can also customize the levels at which SQL Diagnostic Manager provides alert notifications by changing the informational, warning, and critical values either on the Configuration tab of the Alert Configuration window or, for database- and disk-level alerts, on the Database Threshold Configuration window available by selecting the database or disk.

Also, you can [base your alerts on the past performance](#) of the metrics collected by SQL Diagnostic Manager. This is a powerful and effective way to make sure that the alerts you receive are outside of your typical metric ranges.

 For each alert metric, you can enter the specific criteria by double-clicking the values on the Configuration tab.

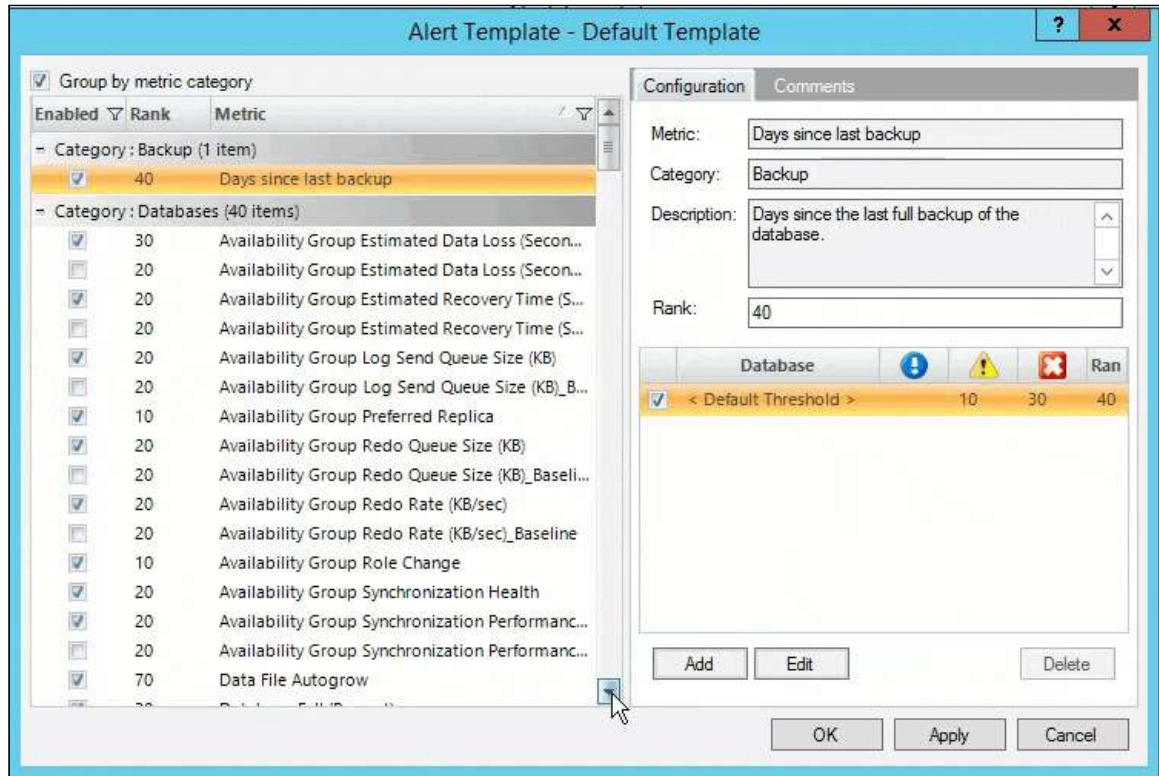
You can [add](#) or [import](#) custom metrics to your alert configuration by using the Add Custom Counter wizard or the Import Custom Counter Wizard on the Administration window. This dialog allows you to [use custom counters to track metrics](#) that SQL Diagnostic Manager begins to monitor. You can select metrics in PerfMon, SQL Server, and even metrics monitored via custom T-SQL script.

To configure the settings for an alert template:

1. Select the metric you want to edit from the list in the Alert Template Configuration window.
2. Check the boxes next to **Informational**, **Warning**, and **Critical** to include alerts for these states.
3. Change the alert thresholds by moving the arrows to the appropriate levels or by double-clicking the value and typing a new threshold level.
4. Select the **Comments** tab and enter information you want displayed in the alert message for this metric.
5. Edit the Rank value if needed.

⚠ The Rank value is a global setting, editing this value affects all instances, and not just one template.

6. Click **Apply**.
7. Click **OK** to accept your changes.



Use alert templates in individual databases or disks within an instance

SQL Diagnostic Manager includes the ability to apply some alert thresholds on a per-database or per-disk basis within an instance. When you create an alert template and include one of these alerts, you cannot select which database or disk to apply the alert to because that is unknown in a template. For that reason, these alerts apply to all databases and disks within the instance to which it is applied. For additional information, see [Configure database thresholds](#) or [Configure disk thresholds](#).

About informational alerts

Informational alerts allow you to set a threshold that, when generated, triggers a status that does not affect the overall status of the server within SQL Diagnostic Manager. You can use informational alerts to notify an administrator of the state of a particular metric for a server or trigger secondary processes that could take action to prevent issue escalation.

⚠ Note that for certain metrics, using the informational alert means that you no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.

Configure database thresholds

The Database Threshold Configuration dialog allows you to set alert thresholds for selected databases on your SQL Server instance. Not all alerts are available for database-specific application. The following alerts include database-specific settings:

- Database Full (Percent)
- Database Full (Size)
- Log Full (Percent)
- Log Full (Size)

Use the **Advanced** button to set [how and when alerts are raised](#) by limiting the number of alerts and notifications generated.

To add a database for specific alert configuration:

1. Right-click a SQL Server instance in the Servers sidebar.
2. Select **Configure Alerts**.
3. Select the database-specific metric you want to edit from the list in the **Alert Configuration** window.
4. On the Configuration tab, click **Add**.
5. Select the database from the available drop-down list.
6. Make the appropriate changes to your alert configuration for this database, and then click **OK**. SQL Diagnostic Manager now displays that database in the list on the Configuration tab.
7. Click **OK** to accept your changes.

To edit the alert threshold configuration for a database:

1. Right-click a SQL Server instance in the Servers sidebar.
2. Select **Configure Alerts**.
3. Select the database-specific metric you want to edit from the list in the **Alert Configuration** window.
4. On the Configuration tab, click the database you want to configure, and then click **Edit**.
5. Change the alert thresholds by moving the arrows to the appropriate levels or by double-clicking the value and typing a new threshold level, and then click **OK**.
6. Click **Apply**.
7. **If you want SQL Diagnostic Manager to replicate your edits to other monitored SQL Server instances, tags, or templates**, click **Yes**.
8. In the Select Modification Targets dialog, select the templates, tags, or instances you want to include, and then click **OK**.
9. Click **OK** to accept your changes.

To delete the alert threshold configuration for a database:

1. Right-click a SQL Server instance in the Servers sidebar.
2. Select **Configure Alerts**.
3. Select the metric from which you want to remove the specific database from the list in the **Alert Configuration** window.
4. On the Configuration tab, click the database you want to delete, and then click **Delete**.
5. Click **Yes** in the confirmation window.
6. Click **Apply**.
7. **If you want to delete the same database from any other SQL Server instances, tags, or templates**, click **Yes**.
8. In the Select Modification Targets dialog, select the templates, tags, or instances you want to apply the same modification, and then click **OK**.
9. Click **OK** to accept your changes.

Configure disk thresholds

The Disk Threshold Configuration dialog allows you to set alert thresholds for selected disks on your SQL Server instance. Not all alerts are available for applying disk-specific thresholds. The following alerts include disk-specific settings:

- OS Disk Free Space (Size)
- OS Disk Full (Percent)

Use the **Advanced** button to set [how and when alerts are raised](#) by limiting the number of alerts and notifications generated.

To add a disk for specific alert configuration:

1. Right-click a SQL Server instance in the Servers sidebar.
2. Select **Configure Alerts**.
3. Select the disk-specific metric you want to edit from the list in the **Alert Configuration** window.
4. On the Configuration tab, click **Add**.
5. Select the disk from the available drop-down list.
6. Make the appropriate changes to your alert configuration for this disk, and then click **OK**. SQL Diagnostic Manager now displays that disk in the list on the Configuration tab.
7. Click **OK** to accept your changes.

To edit the alert threshold configuration for a disk:

1. Right-click a SQL Server instance in the Servers sidebar.
2. Select **Configure Alerts**.
3. Select the disk-specific metric you want to edit from the list in the **Alert Configuration** window.
4. On the Configuration tab, click the disk you want to configure, and then click **Edit**.
5. Change the alert thresholds by moving the arrows to the appropriate levels or by double-clicking the value and typing a new threshold level, and then click **OK**.
6. Click **Apply**.
7. **If you want SQL Diagnostic Manager to replicate your edits to every monitored SQL Server instance**, click **Yes**, select the instances you want to include, and then click **OK**.
8. Click **OK** to accept your changes.

To delete the alert threshold configuration for a disk:

1. Right-click a SQL Server instance in the Servers sidebar.
2. Select **Configure Alerts**.
3. Select the metric from which you want to remove the specific disk from the list in the **Alert Configuration** window.
4. On the Configuration tab, click the disk you want to delete, and then click **Delete**.
5. Click **Yes** in the confirmation window.
6. Click **Apply**.
7. **If you want to delete the same disk from any other SQL Server instance SQL Diagnostic Manager is monitoring**, click **Yes**, select the instances you want to include, and then click **OK**.
8. Click **OK** to accept your changes.

Set advanced alert options

The Advanced Alert Settings window allows you to control how and when alerts are raised by limiting the number of alerts and notifications that are generated.

Alert Duration

[Alert duration](#) allows you to select the total time an alert is raised for an event.

Alert Suppression

[Alert suppression](#) allows you to select the amount of time SQL Diagnostic Manager waits before alerting on a metric that has exceeded threshold limits.

Alert Filters

Alert filters allows you to select:

- Filter out alerts:
 - Minimum Version Store Size that should be reach before an alert is raised.
 - Minimum Version Store Generation Rate at which an alert should be generated.
- [Disk drive exclusions](#) to exclude one or more disk drives from triggering an alert
- [Text and regular expressions](#) to exclude certain words and expressions from triggering an alert

Autogrow Settings

[Autogrow settings](#) allows you to select whether you want SQL Diagnostic Manager to take auto growth into account when calculating the remaining space available.

Database Exclusions

[Database exclusions](#) allows you to select the databases you want to exclude from database alerts.

Job Exclusions

[Job exclusions](#) allows you to select both the jobs and job categories you want to exclude from job alerts.

Sessions Exclusions

[Sessions exclusions](#) allows you to select the applications, host servers, and users you want to exclude from job alerts.


Custom Counters

[Custom counters](#) allows you to select whether SQL Diagnostic Manager generates an alert when a custom counter is not collected.

Configure a custom response when an alert is generated by SQL Diagnostic Manager

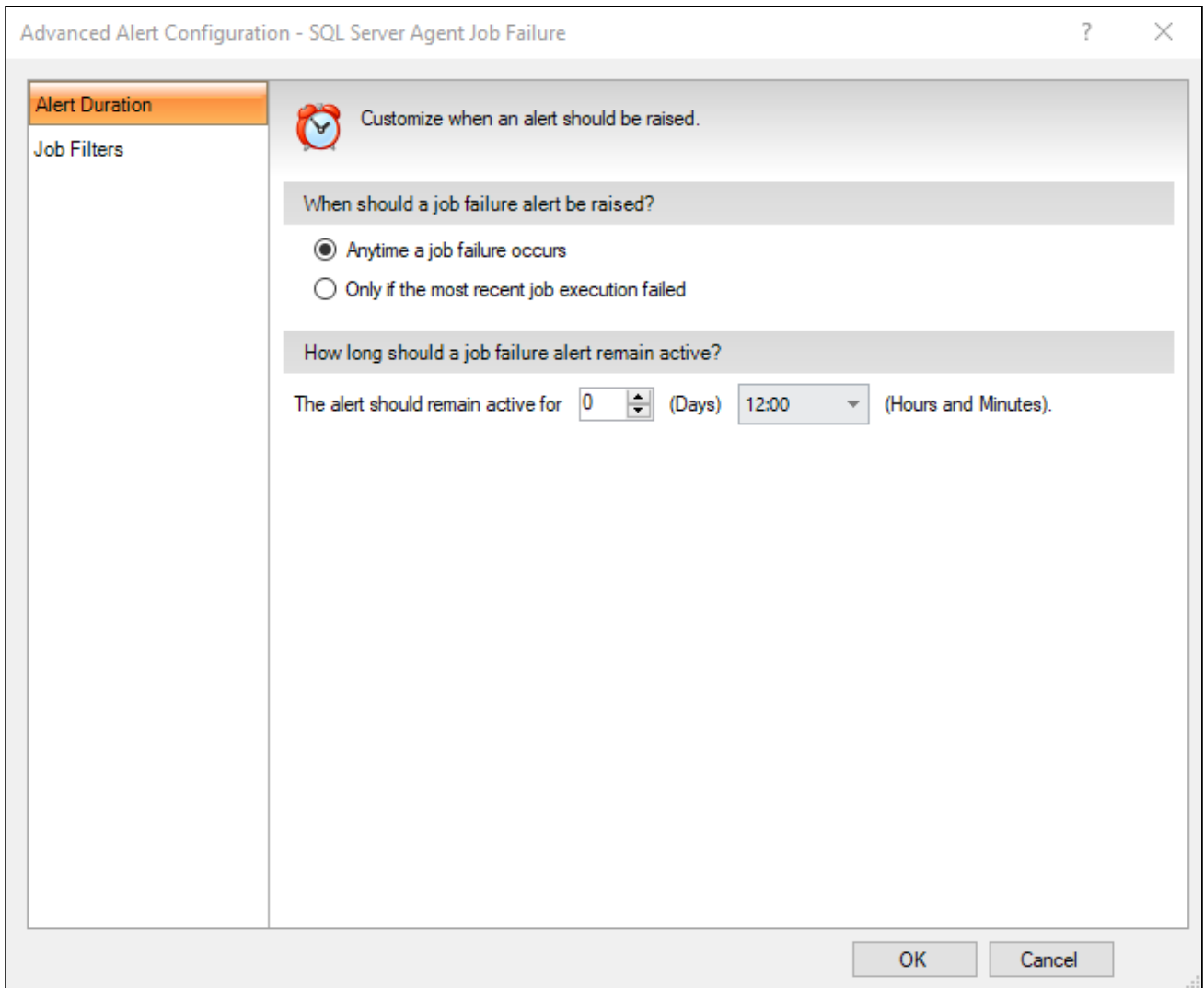
SQL Diagnostic Manager allows you to configure when and how alert responses are generated. You can select from the following alert responses (providers):

- Send an email to a person or group of people
- Log an event in the Windows Event Log
- Enable the Query Monitor
- Execute a program or utility
- Send an event to your Network Management tool
- Execute a SQL Agent Job
- Execute a Script
- Create a To Do (Note that beginning with SQL Diagnostic Manager 7.0, the To Do feature is available only to users who had To Do activated in an earlier 6.x version. This version of SQL Diagnostic Manager does not support this feature for new users.)

To configure a custom response when an alert is generated, select  > **Alert Actions and Responses** from the console toolbar.

Configure alert duration

SQL Diagnostic Manager allows you to determine the total time an alert is raised for an event occurring on a specified SQL Server instance. The Alert Duration tab of the Advanced Alert Configuration window provides a page for you to enter the duration of the alert in hours and minutes. For the SQL Server Agent Job Failure metric, you can specify whether you want SQL Diagnostic Manager to trigger the alert each time any job fails or to raise an alert only upon the failure of the most recently-executed job.



The metrics that allow for specifying alert duration include the following:


- SQL Server Agent Job Completion
- SQL Server Agent Job Failure
- SQL Server Agent Long Running Job (Minutes)
- SQL Server Agent Long Running Job (Percent)

To use ad hoc Alert Configuration to enter how long a job alert remains active:

1. In the Navigation page, click **Servers**.
2. Right-click the appropriate SQL server instance, and then select **Configure Alerts**.
3. Select the metric for which you want to configure an alert.

4. Click **Advanced**.
5. Click **Alert Duration**.
6. Type or use the selection window to enter the alert duration in hours and minutes.
7. Click **OK**.
8. To accept these changes, click **OK**.

To use Alert Configuration templates to enter how long a job alert remains active:

1. Click  > **Alert Configuration Templates**.
2. Click **New** or select an existing template, and click **View/Edit**.
3. *If you clicked New*, type a name, and click **Edit Configuration**.
4. *If you clicked View/Edit*, click **Edit Configuration**.
5. Select the metric for which you want to configure an alert.
6. Click **Advanced**.
7. Click **Alert Duration**.
8. *If you selected the SQL Server Agent Job Completion, SQL Server Agent Long Running Job (Minutes), or SQL Server Agent Long Running Job (Percent) metric*, continue with the next step. *If you selected the SQL Server Agent Job Failure metric*, select **Anytime a job failure occurs** to trigger an alert each time a job failure occurs or select **Only if the most recent job execution failed** if you want to raise an alert only when the most recently-executed job fails.
9. Type or use the selection window to enter the alert duration in hours and minutes.
10. Click **OK**.

Configure alert suppression

In certain environments, it is acceptable for some metrics to exceed the configured alert thresholds for very short periods of time, for example one refresh. Typically these exceptions last less than a minute. If you are monitoring many SQL Server instances or if the amount of data collected is very high, this could produce a significant number of alerts. The Advanced Alert Configuration window allows you to limit the number of these alerts that are generated by letting you input the number of minutes a threshold violation occurs before an alert is raised, reducing the amount of alert "noise" you receive.

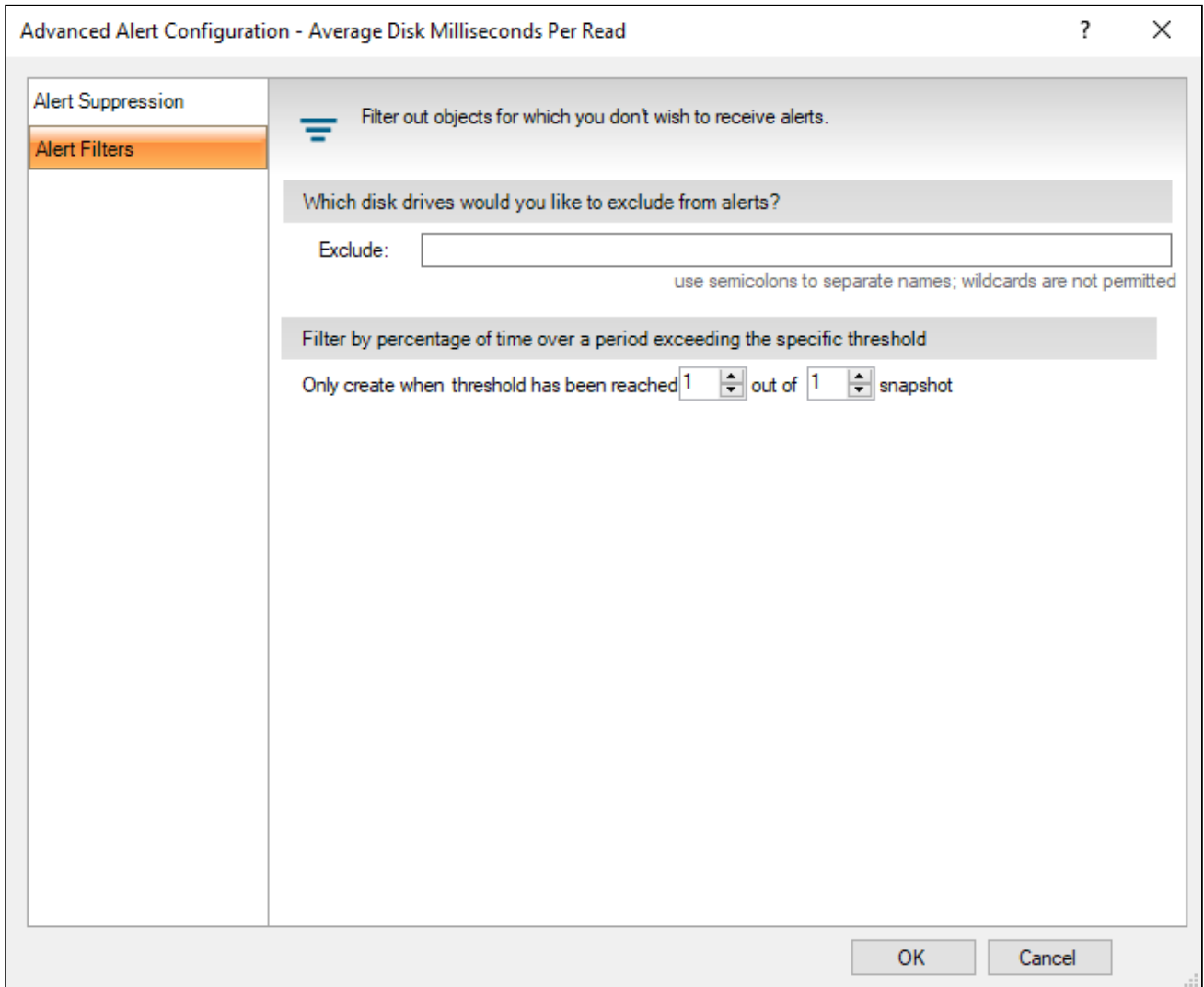
The amount of time you enter here depends on how critical the metric is in your environment and the amount of time SQL Diagnostic Manager waits between refreshes. If you enter a time less than that of the refresh, the second consecutive refresh where the alert is outside the acceptable threshold range prompts the alert.

The following table displays the length of time SQL Diagnostic Manager waits before an alert is raised when the scheduled refresh occurs every six minutes and various values are entered:

Time entered on the Advanced Alert Configuration window	Alert is generated
1-5 minutes	Second Refresh occurs six minutes after the first encounter of the problem
6-11 minutes	Third Refresh occurs 18 minutes after the first encounter of the problem
12-17 minutes	Fourth Refresh occurs 24 minutes after the first encounter of the problem

Configure alert filters - disk drive exclusions

By default, certain metrics include all drives within a specific SQL Server instance when monitoring for events that can trigger an alert. The Alert Filters tab of the Advanced Alert Configuration window allows you to select one of more disk drives to exclude from alerting on the selected metric.




The alerts that allow for disk drive exclusions include the following:

- [Average Disk Milliseconds Per Read](#)
- [Average Disk Milliseconds Per Transfer](#)
- [Average Disk Milliseconds Per Write](#)
- [Disk Reads Per Second](#)
- [Disk Transfers Per Second](#)
- [Disk Writes Per Second](#)
- [OS Average Disk Queue Length Per Disk \(Count\)](#)
- [OS Disk Free Space \(Size\)](#)
- [OS Disk Full \(Percent\)](#)
- [OS Disk Time Per Disk \(Percent\)](#)

To use ad hoc Alert Configuration to exclude disk drives from an alert:

1. Right-click the appropriate SQL server instance, and then select **Configure Alerts** .
2. Select the metric for which you want to configure an alert.
3. Select the appropriate disk, **Edit** , and click **Advanced**.
4. Click the Alert Filters tab.
5. Type one or more disk drives to exclude from alerts.
6. Click **OK** .

To use Alert Configuration Templates to exclude disk drives from an alert:

1. Click  and select **Alert Configuration Templates** .
2. Click **New** or select an existing template, and then click **View/Edit** . When you click **New** , type a name, and click **Edit Configuration** .
3. Click **View/Edit** , click **Edit Configuration**.
4. Select the metric for which you want to configure an alert.
5. Select the appropriate disk, **Edit** , and click **Advanced**.
6. Click the Alert Filters tab.
7. Type the name of the disk drives you want to exclude from alerts. Use semicolons to separate names and do not use wildcard characters.
8. Click **OK**.

Configure text and expression alerts

SQL Server Agent Log and SQL Server Error Log metrics allow users to trigger an alert based on certain text or regular expressions. SQL Diagnostic Manager provides different levels of alerts so you can select whether you want to generate a critical, warning, or informational alert based on your entered data. You can also limit the size of your logs to decrease your response time and avoid using additional space on your server.

Advanced Alert Configuration - SQL Server Agent Log

Alert Filters

Specify text or size limits for which you wish to receive alerts.

What text would you like to raise alerts for?

Critical:

Warning:

Informational:

use semicolons to separate strings; use % for wildcards

What regular expressions would you like to raise alerts for?

Critical:

Warning:

Informational:

one regular expression per line

What size would you like to limit your log to?

Size (MB):


Filter by percentage of time over a period exceeding the specific threshold

OK Cancel

To use ad hoc Alert Configuration to enter text or a regular expression to trigger an alert:


1. Right-click the appropriate SQL server instance, and then select **Configure Alerts**.
2. Select the metric for which you want to configure an alert.
3. Click **Advanced**.
4. Under **What text would you like to raise alerts for?** Type the text in the appropriate field. Use semicolons to separate names and a percent (%) character as a wildcard.
5. Under **What regular expressions would you like to raise alerts for?** Type the expression in the appropriate field. Type only one regular expression per line.
6. Click **OK**.


To use Alert Configuration Templates to enter text or a regular expression to trigger an alert:

1. Click  and select **Alert Configuration Templates**.
2. Click **New** or select an existing template, and then click **View/Edit**.
3. **If you clicked New**, type a name, and then click **Edit Configuration**.
4. **If you clicked View/Edit**, click **Edit Configuration**.
5. Select the metric for which you want to configure an alert.
6. Click **Advanced**.
7. Under **What text would you like to raise alerts for?** Type the text in the appropriate field. Use semicolons to separate names and a percent (%) character as a wildcard.
8. Under **What regular expressions would you like to raise alerts for?** Type the expression in the appropriate field. Type only one regular expression per line.
9. Click **OK**.

Limit the size of a log that SQL Diagnostic Manager reads

In an attempt to improve response time and avoid using unnecessary space on your server, you can designate the maximum size of your SQL Server Agent log and your SQL Server Error log that SQL Diagnostic Manager reads. When a log reaches the size limit set in the Alert Configuration, it triggers an alert. SQL Diagnostic Manager does not display in the Logs view any data from error or agent logs in excess of the size limit. A quick solution to decrease the size of your file is to click Cycle Server Log to [archive the information in the most recent log and start a new log](#). You can also cycle the log using SSMS.

 Before cycling your log, it is important to note that if you have a limit set, then the log file deleted is the oldest log file stored. If you want to keep that log file, copy the file elsewhere before cycling the log.

 SQL Server 2000 users also must restart the SQL Server Agent Service after cycling the SQL Server Agent log. The only way to cycle the agent log on SQL Server is to cycle the agent service. This process results in a new log.

To adjust the log size limit that SQL Diagnostic Manager reads:

1. View the Alert Configuration for your monitored SQL Server instance.
2. Select either the **SQL Server Agent Log** or the **SQL Server Error Log** alert, and then click **Advanced**.
3. In the Alert Filters tab, adjust the **Size (MB)** field based on what you want for your log size limit.
4. Click **OK**, and then click **Apply**.
5. Click **OK** in the confirmation message if you want to apply your change to any other SQL Server instances, or click **No**.

Configure size remaining

SQL Server can automatically expand a database or log when necessary based on a set of parameters. The Auto-grow Settings tab of the Advanced Alert Configuration window allows you to specify whether you want to take this autogrowth into account when calculating the remaining space available.

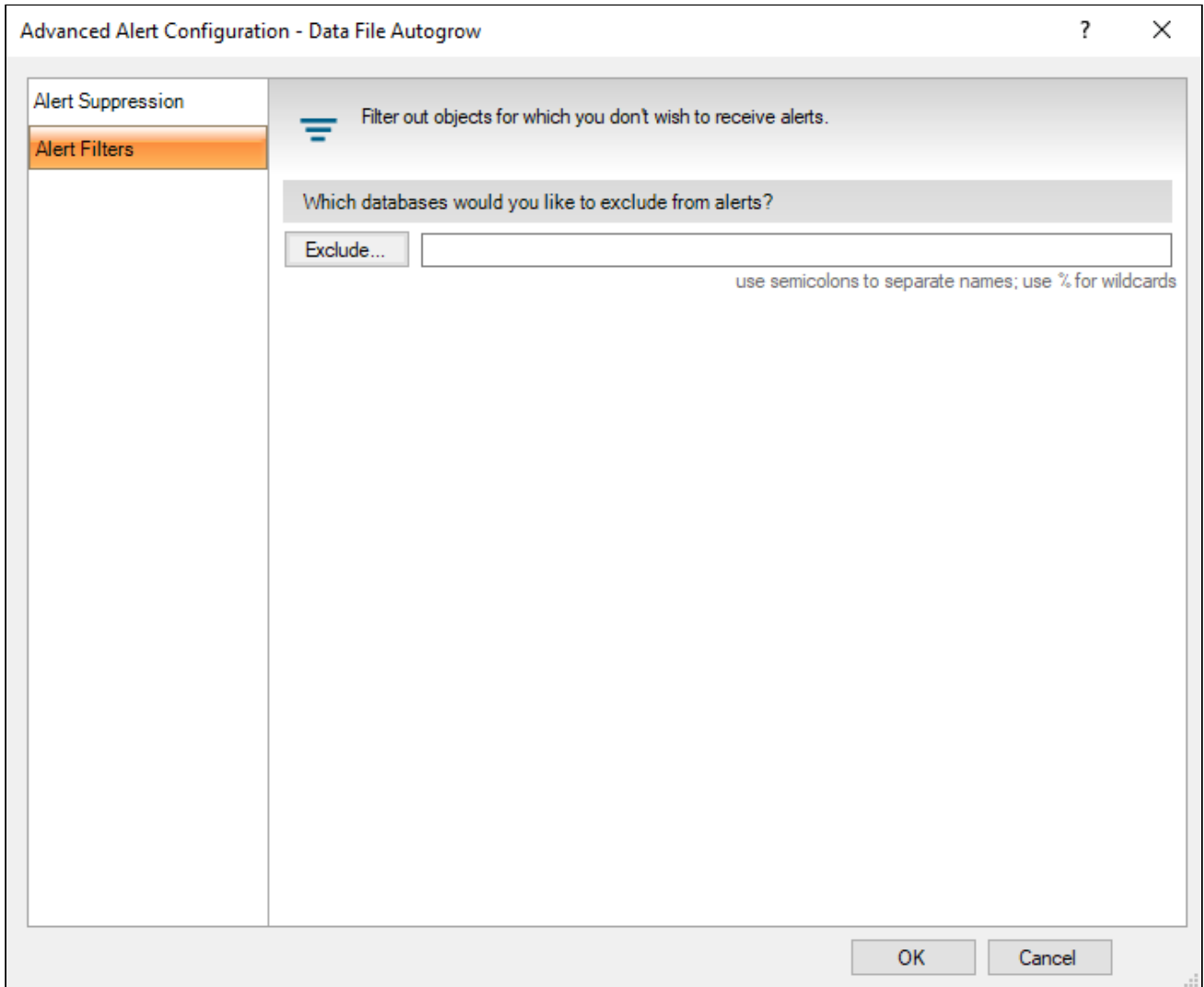
SQL Diagnostic Manager allows you to select whether to trigger an alert based on the current database or log used size divided by the maximum possible size. Click **Yes, alert on the current used size divided by the maximum possible size** on the Autogrow Settings page to set the alert to include auto grow or click **No, alert on the current used size divided by the current file size** if you do not want to take SQL Server auto-growth into consideration.

Understanding the downside of Auto-grow

You can avoid unnecessary long reads and poor response time from SQL Server by minimizing the number of auto-grow events occurring in your environment. Each time an auto-growth event occurs, SQL Server holds up database processing while looking for disk space for the additional data. Usually this new space is not adjacent to the existing space and causes physical disk fragmentation. As more and more auto-growth events occur, the more the disk becomes fragmented, leading to longer periods while SQL Server reads the databases.

Configure alert filters - database exclusions

The Alert Filters tab of the Advanced Alert Configuration window allows you to select databases to exclude from alerting on the selected metric. You can limit the amount of alert noise generated for very small databases, which tend to report a high fragmentation rate, or databases that are in single user mode and are usually in use.



The alerts that allow for database exclusions include the following:

- [Data File Autogrow](#)
- [Database Full \(Percent\)](#)
- [Database Full \(Size\)](#)
- [Database Status](#)
- [Days since last backup](#)
- [Filegroup Space Full \(Percent\)](#)
- [Filegroup Space Full \(Size\)](#)
- [Log File Autogrow](#)
- [Log Full \(Percent\)](#)
- [Log Full \(Size\)](#)
- [Table Fragmentation \(Percent\)](#)

Access the Alert Filters tab of the Alert Configuration window

1. Right-click the SQL Server instance you want to manage and then select **Configure Alerts**.
2. Select the alert you want to configure.
3. **If the alert is database-specific**, select the appropriate database, click **Edit**, and select **Advanced**. If the alert is not database-specific, click **Advanced**.
4. Click the Alert Filters tab.

Exclude a database from an alert

You can select one or more databases to exclude from alerting on the selected metric.

1. Access the Alert Filters tab using the previous steps.
2. Click **Exclude**.
3. Select one or more databases from the list and click **Exclude**.
4. Click **OK**.
5. To accept these changes, click **OK**.

Configure job exclusions

The Alert Filters tab of the Advanced Alert Configuration window allows you to select jobs or job categories to include or exclude from alerting for the specified metric. Filtering allows you to limit the amount of alert noise generated for SQL Server Agent Jobs or Job categories that, for example, are known to run for a long time.

- ✓ By default, SQL Diagnostic Manager alerts on all Jobs. If you check the **Alert on individual job steps** box, all job steps generate alerts. Using the Include filter limits job alerts to only those SQL Server Agent Job steps that apply to the filter. The Exclude filter limits the job alerts to every SQL Server Agent Job step except for those included in the Exclude filter.

Advanced Alert Configuration - SQL Server Agent Job Failure

Filter out jobs or job steps you do not want to receive alerts for

Should individual job steps be evaluated for alert conditions?

Alert on individual job steps

Include

Operator	Category	Operator	Job Name

Add Edit Remove

Exclude

Operator	Category	Operator	Job Name

Add Edit Remove

OK Cancel

The metrics that allow for SQL Server Agent Job exclusions include the following:

- [SQL Server Agent Job Completion](#)
- [SQL Server Agent Job Failure](#)
- [SQL Server Agent Long Running Job \(Minutes\)](#)
- [SQL Server Agent Long Running Job \(Percent\)](#)

To filter SQL Server Agent Job alerts:

1. Open the Alert Configuration window and select the alert you want to configure.
2. Click **Advanced**.
3. Select the **Job Filters** tab.
4. Check the **Alert on individual job steps** box if you would like an alert for each step in each job.
5. Click the **Add** button under the Include or Exclude filters.
6. Enter the **Job Name** or **Job Categories** to include or exclude and click **OK**. Use the percent symbol (%) as a wildcard.
7. Click **OK** to apply your changes.

You can set thresholds for the SQL Server Agent Long Running Job (Minutes) and SQL Server Agent Long Running Job (Percent) metrics on your SQL Server instance.

To add a job for specific alert configuration:

1. Right-click a SQL Server instance in the Servers sidebar.
2. Select **Configure Alerts**.
3. Select the job-specific metric you want to edit from the list in the Alert Configuration window.
4. On the Configuration tab, click **Add**.
5. Specify or select the job name.
6. Make the appropriate changes to your alert configuration for this job, and click **OK**. SQL Diagnostic Manager now displays the job in the list on the Configuration tab.

Follow these steps to edit the alert threshold configuration for a job:

1. Right-click a SQL Server instance in the Servers sidebar.
2. Select **Configure Alerts**.
3. Select the job-specific metric you want to edit from the list in the Alert Configuration window.
4. On the Configuration tab, select the job you want to configure, and click **Edit**.
5. Change the alerts thresholds by moving the arrows to the appropriate level or by entering new values of the threshold level on the Job Threshold Configuration window.
6. Click **OK**.

Configure session exclusions

The Advanced Alert Configuration window allows you to exclude applications, host servers, and users from the selected alert. You can limit the amount of alert noise generated by known applications, power users, or even a whole list of power users, such as a development team, that commonly exceed your alert thresholds.

The metrics that allow for session exclusions include the following:

- [Blocking Session Wait Time \(Seconds\)](#)
- [Oldest Open Transaction \(Minutes\)](#)
- [Session CPU Time \(Seconds\)](#)

To configure session exclusions:

1. Open the Alert Configuration window and select the alert you want to configure.
2. Click **Advanced**.
3. Enter the applications, host servers, and users you want to exclude from the alert using % for wildcards and semicolons to separate application names.
4. Click **OK** to save your configuration.

Configure custom counter suppression

The Advanced Alert Configuration window allows you to select whether you want an alert created when SQL Diagnostic Manager cannot collect a custom counter. Preventing an alert when you cannot collect the custom counter eliminates any potential noise the alert could generate.

- ✔ If a custom counter collection fails, it is included in a critical summary alert that lists all the custom counters that it could not collect.

Add or edit job filter

The Add Job Filter or Edit Job Filter window allows you to create or edit a filter to determine when you want a SQL Server Agent job condition to raise an alert.

Access the add or edit job filters

To access the Add Job Filter or Edit Job Filter window:

1. Right-click a SQL Server instance in the Servers tree.
 2. Select **Configure Alerts**.
 3. Select one of the following alerts:
 - SQL Server Agent Job Completion
 - SQL Server Agent Job Failure
 - SQL Server Agent Long Running Job (Minutes)
 - SQL Server Agent Long Running Job (Percent)
- Click **Advanced**, and select **Job Filters**.
 - Click **Add** or **Edit** in either the Include or Exclude list.

Use the Job Filters tab

The Job Filters tab allows you to include or exclude a job by job category, name, or step name if you check **Alert on individual job steps**.

To add a job filter:

1. Select **Like** or **Equals** in the Operator drop-down window for either the Job Category, Job Name, or Step Name field.
2. Enter the value to filter the job for in the Value field or click the **Browse** button to browse for the object you want to filter.





✔ Use the percent symbol (%) as a Wildcard.

3. Click **OK**.

⚠ The Browse button is only available when adding or editing job step filters for a single SQL Server instance.

Alert value configuration

The Value Configuration window allows you to select whether you receive an alert for a particular event based on the selected level for that event. For example, if you select the Critical alert level for **Offline** for Database Status, you receive a Critical Level alert if the Database is ever offline. The alerts are available as shown in the following image.

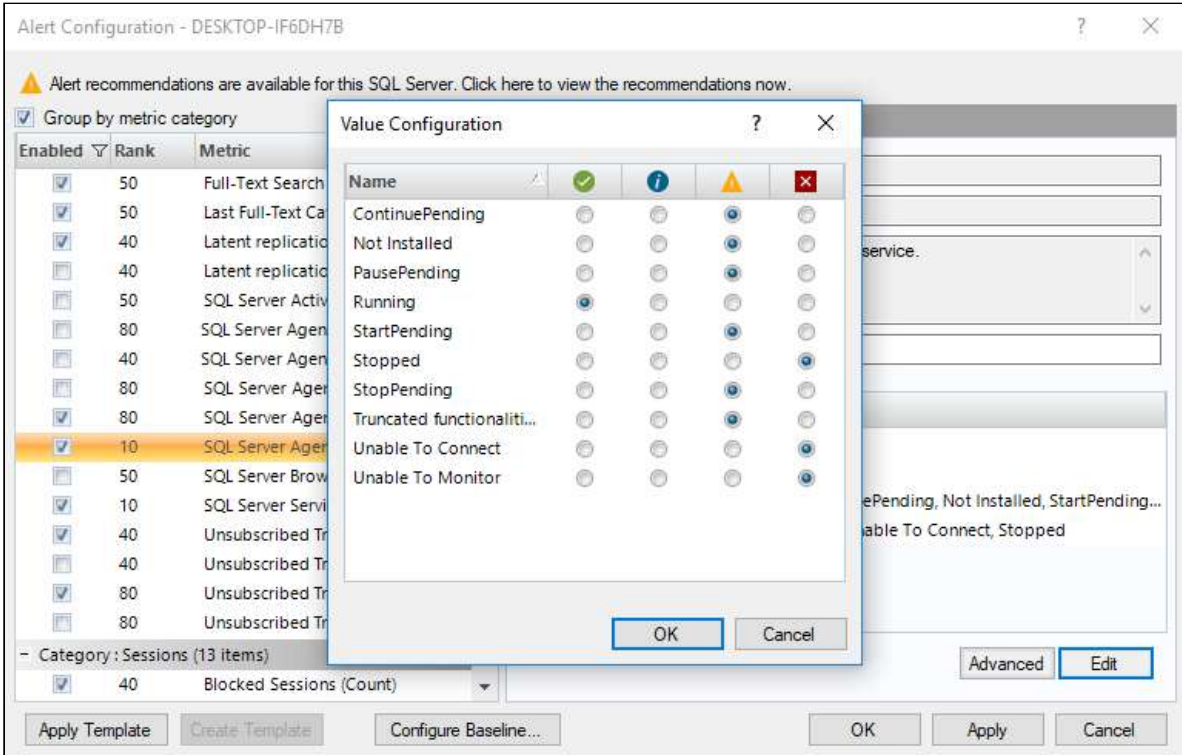
Alert Level	Description
	The selected metric is in an OK state.
	The selected metric is in an Informational state.
	The selected metric is in a Warning state.
	The selected metric is in a Critical state.

Access the Value Configuration window

SQL Diagnostic Manager displays the Value Configuration window when you attempt to edit values for certain metrics.

To access the Value Configuration window:

1. Right-click a monitored SQL Server instance, and then select **Configure Alerts**.
2. In the Alert configuration window, select one of the following metrics:
 - Database Status
 - Mirroring Status
 - OS Metrics Collection Status
 - DTC Status
 - Full-Text Search Status
 - SQL Server Agent Status
 - SQL Server Status
 - SQL Server Agent Job Completion
 - Host Power State
 - VM Power State
3. Click **Edit**.



About informational alerts

Informational alerts allow you to set a threshold that, when generated, triggers a status that does not affect the overall status of the server within SQL Diagnostic Manager. You can use informational alerts to notify an administrator of the state of a particular metric for a server or trigger secondary processes that could take action to prevent issue escalation.

⚠ Note that for certain metrics, using the informational alert means that you no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.

Configure alerts for important cluster events

SQL Diagnostic Manager allows you to monitor your clustered SQL Server environment including alerts when a failover occurs and when the preferred active cluster node changes.

An important way to track clustering events using SQL Diagnostic Manager is through the usage of cluster alerts.

Cluster Failover Alert

By default, this alert is enabled and set to Critical. When a failover occurs, SQL Diagnostic Manager generates an alert and then performs one of the following actions:

- Clears after 12 hours if no additional alerts are generated on the cluster.
- Generates an additional alert if another cluster failover occurs.
- Clears instantly when manually cleared.

Cluster Active Node Alert

The preferred node. You can change this setting on the [Cluster Settings](#) tab of the Monitored SQL Server Properties dialog box. The Cluster Active Node is triggered when SQL Diagnostic Manager detects that a clustered SQL Server instance is running on a node other than the configured preferred node.

Best practices for using the Cluster Alerts

Choose the alert that best reflects your environment and setup notifications for it. Disable the other alert if it is not necessary.


Configure how SQL Diagnostic Manager responds to alerts

The Alert Actions and Responses window allows you add, modify, import/export, and configure alert responses. You can:

- [Configure your email \(SMTP\) settings](#) to send out alert notifications using the email server available on your network.
- [Configure your network management \(SNMP\) settings](#) to set up a network management alert response.

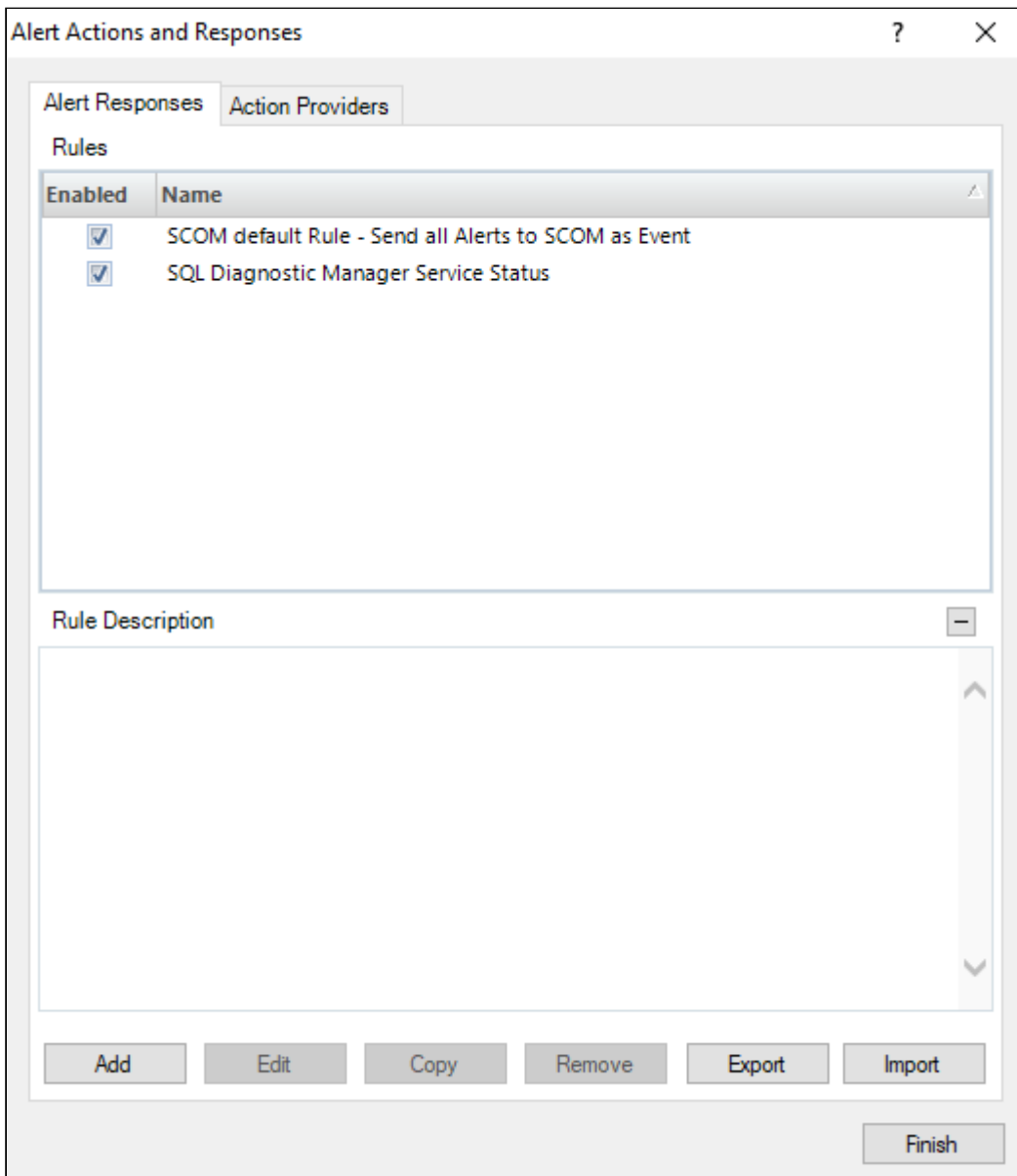
✓ When you first add your SQL Server instance to SQL Diagnostic Manager, you should begin setting up your alert thresholds, and then set up your alert action rules and providers to specify who receives related notifications.

Access the Alert Actions and Responses window

You can open the Alert Actions and Responses window from the SQL Diagnostic Manager Console menu by selecting  > **Alert Actions and Responses**.

Available Alert Responses tab actions

You can enable or disable an action provider by checking/clearing the box in the *Enabled* column.



i SQL Diagnostic Manager provides two rules enabled by default. You can disable, edit, copy, or remove them.

Add

Click **Add** to [create a new alert response](#).

Edit

Click **Edit** to make modifications to an existing alert response.

Copy

Click **Copy** to make a duplicate of an existing alert response. The copy function can save time when you want to change just one setting on a particular rule.

Remove

Click **Remove** to delete the selected alert response.

Export

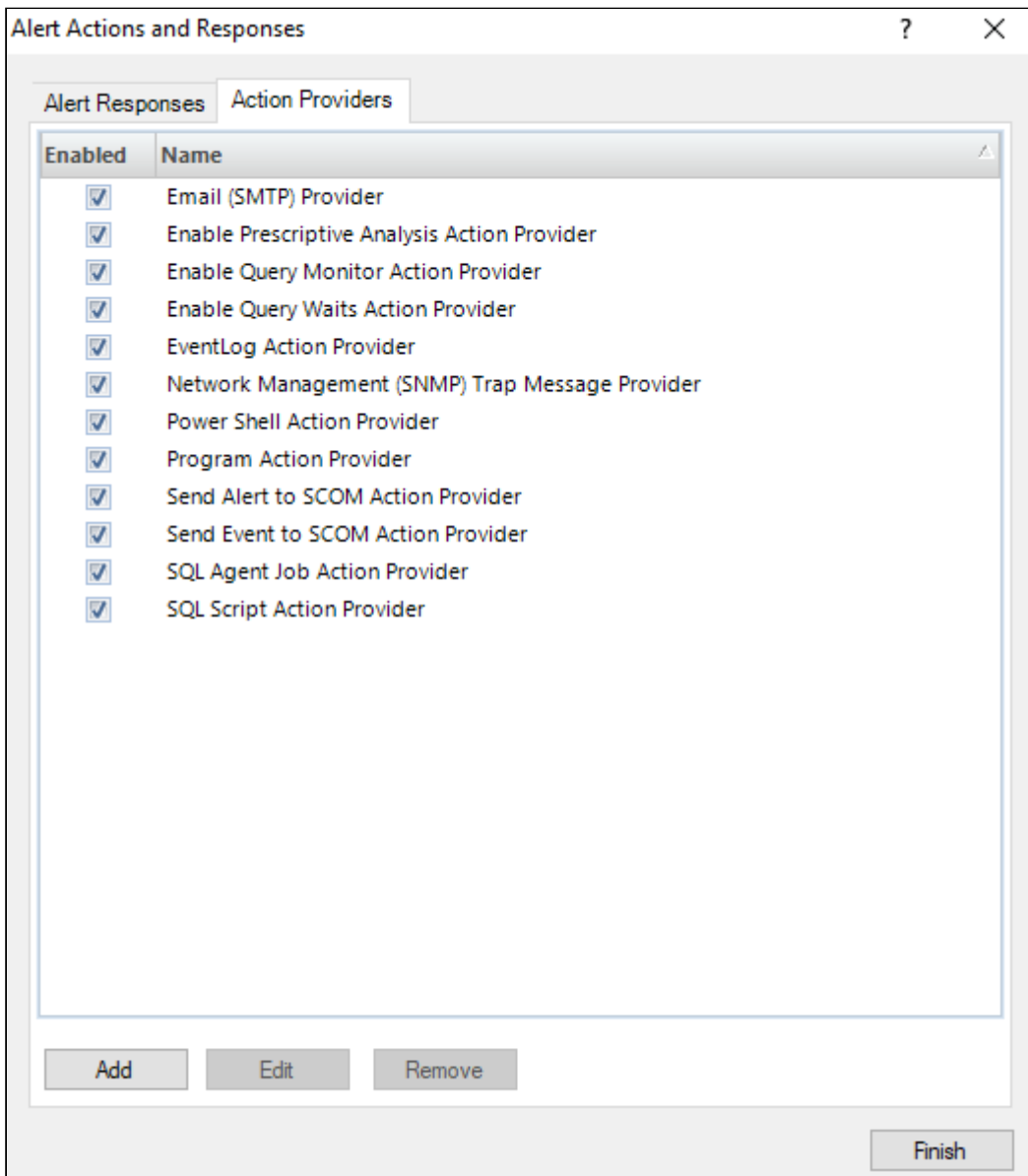
Click **Export** to export an alert response XML file.

Import

Click **Import** to import an alert response XML file.

Available Action Providers tab actions

You can enable or disable an action provider by checking/clearing the box in the *Enabled* column.



Add

Click **Add** to open the [Alert Communications Wizard](#) and create a new SMTP or SNMP action provider.

Edit

Click **Edit** to edit an SMTP or SNMP action provider.

Remove

Click **Remove** to delete an SMTP or SNMP action provider.

Configure automated responses to alerts

Use the **Alert Response** window to create or modify your new alert response rule.

Access the Alert Response window

1. Select  > **Alert Actions and Responses**.

Add a new rule

1. Click **Add** on the **Alert Responses** tab of the Alert Actions and Responses dialog box.
2. Enter a **Name** for the rule.
3. Check the boxes for the conditions and destinations for the rule.
4. Click **Enable** to activate the associated action. Note that all actions are disabled by default.
5. If you enabled an action, provide any required information for the selected action.
6. Click **Add** to select additional actions to perform in response to the alert. This feature allows you to add multiple instances of certain actions.
7. Click the hyperlink text to select specific conditions for the rule.
8. Click **OK** when finished.

Available conditions

By default, alert response rules are not limited to specific SQL Server instances, metrics, severity levels, and times. You can use conditions to tailor the triggers of your alert responses. The following conditions help you manage alert response rules.

Where the SQL Server Instance is in *specified list*

Trigger a response when the listed SQL Server instance is included in the displayed list. Click **specified list**, and then select one or more instances to include in the alert response trigger. Click **Select all** to include all of the displayed instances.

Where SQL Server Instance has a tag in *specified list*

Trigger a response when the listed SQL Server instance contains a tag in the displayed list. Click **specified list**, and then select one or more tags to include in the alert response trigger. Click **Select all** to include all of the displayed tags.

Where the metric is in the *specified list*

Trigger a response if the metric is included in the displayed list. Click **metric**, and then select one or more metrics to include in the alert response trigger. Click **Select all** to include all of the displayed metrics.

Where all metrics are in the *specified list*

Trigger a response only if multiple metrics in the specified list reach the specified threshold(s). Click **metric**, and then select metrics to include in the alert response trigger. Click **Select all** to include all of the displayed metrics.

Where metric severity has changed

Trigger a response when the metric severity changes regardless of level. Click the displayed severity level, and then select one or more severity levels to include in the alert response trigger.

Where refresh occurred during a *specific time frame*

Trigger a response if the refresh occurred during the displayed time frame. Click **during this time**, and then select the time and date range to set the alert response trigger time frame.

Available action providers

Action providers allow you to select actions that you want to occur as part of the alert response. By default, all actions are disabled. To enable an action, click **Enable** next to the appropriate action. The following actions help you manage alert response rules:

- [Prescriptive Analyzer Action Provider](#)
- [Query Monitor Action Provider](#)
- [SCOM Alert Action Provider](#)
- [EventLog Action Provider](#)
- [Network Management Protocol \(SNMP\) Trap Message Provider](#)
- [Program Action Provider](#)
- [Email \(SMTP\) Provider](#)
- [SCOM Event Action Provider](#)
- [SQL Agent Job Action Provider](#)
- [SQL Script Action Provider](#)
- [Query Waits Action Provider](#)
- [PowerShell Action Provider](#)

Prescriptive Analyzer Action Provider

When the criteria for the new alert rule is met, it triggers the Prescriptive Analyzer to run a prescriptive analysis on a specific server based on selected performance categories.

To enable the prescriptive analyzer as an alert response:

1. Enable the Prescriptive Analyzer provider, and the Prescriptive Analysis Settings window displays.
2. Select the [performance categories](#) of your choice.
3. Click **OK**.

Query Monitor Action Provider

When the criteria for the new alert rule is met, it triggers the Query Monitor to start so that you can collect more information.

To enable query monitor as an alert response:

1. Enable the Query Monitor provider and the Query Monitor Settings dialog displays.
2. Select either **Enable query monitor** to enable the query monitor until it is manually disabled or **Enable query monitor for a limited time**.
3. Click **OK**.

SCOM Alert Action Provider

When the criteria for the new alert rule is met, SQL Diagnostic Manager sends an alert to SCOM.

To create SCOM alert response:

1. Send **to SCOM as Alert** option.
2. Select the metrics you want to be sent to SCOM as alerts. You can also select all available metrics.
3. Click **OK**.

EventLog Action Provider

When the criteria for the new alert rule is met, an entry to the EventLog is created.

Network Management Protocol (SNMP) Trap Message Provider

When the criteria for the new alert rule is met, SQL Diagnostic Manager sends an SNMP Trap message to the network management console based on the [configuration of your email settings](#) and [network management settings](#).

Program Action Provider

When the criteria for the new alert rule is met, you can specify a program for the SQLdm Management Service to run in response to the alert.

To start a program as an alert response:

1. Enable the **Program Action** provider and the Program Action dialog displays.
2. Enter a description for the **Program** action.
3. Enter the full path to the program you want to launch as an alert response. Use the **More Options** drop-down menu to select from common runtime arguments.

✔ You can run executable (.exe) files only through the Program Action Provider. If you need to run a batch (.bat) file, use `cmd /c run.bat`.

4. Enter the full path of the directory in which you want the program to run.

✔ The **Start in** field is useful when you want a different directory for where an action begins and where the program resides.

5. Click **Test** to make sure that the SQLdm Management Service can run the program.
6. Click **OK**.

Email (SMTP) Provider

When the criteria for the new alert rule is met, SQL Diagnostic Manager sends an alert email message to the SMTP server, which then forwards the message to the specified email address. [Configure your email settings](#) and [network manager settings](#) to receive these alert email messages.

SCOM Event Action Provider

When the criteria for the new alert rule is met, SQL Diagnostic Manager sends an event to SCOM.

To create SCOM alert response:

1. Send **to SCOM as Event** option.
2. Select the metrics you want to be sent to SCOM as alerts. You can also select all available metrics.
3. Click **OK**.

SQL Agent Job Action Provider

When the criteria for the new alert rule is met, you can specify a SQL Agent Job to run in response to the alert.

To start a SQL Agent Job as an alert response:

1. Enable the SQL Agent Job Action provider and the SQL Agent Job Action dialog displays.
2. Select the SQL Server instance from the **Server** drop-down list or select **%(Instance)** to run the SQL Agent Job on the server that caused the alert.
3. Click **Browse** next to the **Job Name** field.

4. On the Select Job window, select the SQL Server instance the SQL Agent Job is located on and click **Load**.
5. Select the SQL Agent job from the list and click **OK**.
6. If you want to select a particular Job Step to run, click the **Browse** button in the Job Step window
 - a. In the Select Job Step window click **Load**.
 - b. Select the Job step from the list and click **OK**.
 - c. Click **Test** to verify your configuration.
 - d. Click **OK**.

SQL Script Action Provider

When the criteria for the new alert rule is met, you can specify a SQL Server script to run in response to the alert.

To run a SQL Script Action Provider as an alert response:

1. Enable the SQL Script Action provider and the SQL Script Action dialog displays.
2. Enter a description for the SQL Script Action in the Description field.
3. Click **Browse** to select the SQL Server instance or leave it at the default of **%(Instance)** to run the SQL Script on the server that caused the alert.
4. Either type in the T-SQL you want to run when the alert criteria is met, or paste in the T-SQL from another application.
5. Click **Test** to verify your SQL script.
6. Click **OK**.

Query Waits Action Provider

When the criteria for the new alert rule is met, it triggers the Query Waits to start so that you can collect more information.

To enable query waits as an alert response:


1. Enable the Query Waits provider and the Query Waits Settings dialog displays.
2. Select either **Enable query waits** to enable the query monitor until it is manually disabled or **Enable query waits for a limited time** and configure the duration.
3. Click **OK**.

PowerShell Action Provider

When the criteria for the new alert rule is met, you can specify a PowerShell script to run in response to the alert.

To run a PowerShell script as an alert response:

1. Enable the PowerShell Action provider and the PowerShell Action dialog displays.
2. Enter a description for the PowerShell command in the Description field.
3. Either type in the PowerShell command you want to run when the alert criteria is met, or paste in the PowerShell command from another application.
4. Click **Test** to verify your PowerShell command.
5. Click **OK**.

 After you create a new rule, you can then create a To-Do item and assign it to specific users for when the conditions configured in your notification rule are met. Note that beginning with SQL Diagnostic Manager 7.0, the To-Do feature is available only to users who had To Do activated in an earlier 6.x version. This version of SQL Diagnostic Manager does not support this feature for new users.

Change the condition value

You can edit the rule description by clicking the underlined value from the **Step 3** section. SQL Diagnostic Manager displays a list of currently defined items for you to select as the new value. Make the necessary updates, and then click **OK**.

Alert Response

Name:

Step 1: Add condition(s)

- Where SQL Server Instance has a tag in specified list
- Where metric is in specified list
- Where all metrics are in specified list
- Where metric severity has changed
- Where refresh occurred during a specific time frame

Step 2: Select action(s)

Enable	Enable the Prescriptive Analyzer to analyze server for selected categories .	Remove
Enable	Enable the Query Monitor for this time period	Remove
Enable	Enable the Query Waits for this time period	Remove
Enable	Write an event to the Windows Event Log	Remove
Enable	Run the following SQL Agent Job: SQL Agent Job	Remove
Enable	Run the following Power Shell Script: Power Shell Script	Remove
Enable	Run the following program: Program	Remove

Step 3: Edit the rule description (click an underlined value)

For each scheduled refresh occurring [during this time](#)
 Where SQL Server Instance has tags ['Production Servers'](#)
 and any metric severity is [Informational or Warning or Critical](#)
 and metric severity has changed
 send an email to dbaoncall@mycompany.com

Add action provider

The Add Action Provider dialog box allows you to select additional action providers to use for your alert response. The list displays the action providers that support simultaneous responses. The following choices are available:

Email (SMTP) Provider

When the criteria for the new alert rule is met, SQL Diagnostic Manager sends an alert email message to the SMTP server, which then forwards the message to the specified email address. SQL Diagnostic Manager allows you to [configure your email settings](#) and [configure your network management settings](#) to properly route alert messages.

PowerShell Action Provider

Execute a PowerShell command when the criteria of your Alert Response rule is met.

Program Action Provider

Start a program when the criteria of your Alert Response rule is met.

SQL Agent Job Action Provider

Execute a SQL Agent job when the criteria of your Alert Response rule is met.


SQL Script Action Provider

Execute a SQL Script when the criteria of your Alert Response rule is met.

Alert Communications wizard

The Alert Communications wizard allows you to set up and configure SMTP (email) and SNMP (network management) providers for your alerts. Click **Next** to continue to the **Choose Provider Type** window of the wizard.

Access the Alert Communication wizard

1. Select  > **Alert Actions and Responses** from the SQL Diagnostic Manager toolbar menu.
2. Select the Action Providers tab and click **Add**.

Choose provider type

The Choose Provider Type window of the Alert Communication wizard allows you to choose which type of provider you want to create. Select either Simple Mail Transfer Protocol (SMTP) or Simple Network Management Protocol (SNMP) from the Provider Type drop-down, give the Provider a name and click **Next** to configure your provider.


Simple Mail Transfer Protocol (SMTP)

This provider allows you to send an email to one or multiple email addresses whenever alerts are generated.

Simple Network Management Protocol (SNMP)

This provider allows you to send an event to your Network Management system whenever alerts are generated.

Access the Choose Provider Type window

1. Select  > **Alert Actions and Responses** from the SQLDM toolbar menu.
2. Select the Action Providers tab and click **Add**.
3. Click **Next** on the Welcome to the Alert Communications wizard window.

Configure email settings

The [Alert Communications wizard](#) allows you to configure email settings for sending out alert notifications using the email server available on your network.

Configure your email settings

To configure your email settings:


1. Click **Alerts** in the Navigation pane, and select **Alert Actions and Responses**.
2. Click the Action Providers tab.
3. Click **Add**.
4. On the Welcome page of the Alert Communications wizard, click **Next**.
5. Select the **Simple Mail Transfer Protocol (SMTP)** provider type, type a unique name, and then click **Next**.
6. Type the SMTP address, port number, and the number of seconds to wait before a timeout occurs.
7. **If the Server requires authentication**, check **Server requires authentication**, and then type the appropriate login information for the email server.
8. Type the **Name** and **E-mail** address you want to appear in the **From** field in alert notifications.
9. To verify the connection to the SMTP server, click **Test**. Type the email address to send the test email, and then click **OK**.
10. Verify that the correct recipient received the test email message, and then click **OK**. **If the test is unsuccessful**, review your settings.
11. Click **Finish**.

Configure network management settings


The [Alert Communications wizard](#) allows you to set up Simple Network Management Protocol (SNMP) for sending out alert events to your Network Management system on the SNMP Configuration window.

SQL Diagnostic Manager includes a MIB that you can use in your Network Management System to format the events that are sent by SQL Diagnostic Manager.

The MIB file is located in the root directory of your SQL Diagnostic Manager installation and is named [IDERA MIB](#).


 SQL Diagnostic Manager supports SNMPv1 and SNMPv2.

Access the Alert Communications wizard SNMP Configuration window

1. Select  > **Alert Actions and Responses** from the SQL Diagnostic Manager toolbar menu.
2. Select the Action Providers tab and click **Add**.
3. Click **Next** on the Welcome to the Alert Communications wizard window.
4. Select **Simple Network Management Protocol (SNMP)** from the **Provider Type** drop-down.
5. Enter a name for your provider.
6. Click **Next**.

Set up your SNMP communication

1. Enter the SNMP address, port number, and the community name that you want to use.
2. Click **OK**.

 It is best to specify a computer name for the address, especially in a DHCP environment. If the manager has a static address, enter the IPv4 formatted address instead.

3. Click **Test** to verify the connection to the SNMP enabled management console.
4. Verify that the test event arrived and click **OK** if the test is successful.
5. Click **Finish**.

Understanding the IDERA MIB

A MIB is a standardized file that defines the data that is sent to the Network Management Console in the SNMP Trap events generated by SQL Diagnostic Manager.

SNMP traps enable an agent to spontaneously inform the management station of significant events by way of an unsolicited SNMP message. Traps are suitable when there is a large number of devices and each device has a large number of objects.

Note that in order for a management system to interpret a trap sent to it by an agent, it needs to have the MIB for that trap loaded.


The MIB file (Idera.MIB) defines the following data:

Name	OID	Syntax	Description
Alert Summary	.1.3.6.1.4.1.24117.1.2.1	DisplayString	Summary text of the alert
Alert Text	.1.3.6.1.4.1.24117.1.2.2	DisplayString	Full text description of the alert
Metric	.1.3.6.1.4.1.24117.1.2.3	DisplayString	SQLdm metric that triggered the alert
Severity	.1.3.6.1.4.1.24117.1.2.4	DisplayString	Severity of the alert
Instance	.1.3.6.1.4.1.24117.1.2.5	DisplayString	The SQL server instance that triggered the alert
Database	.1.3.6.1.4.1.24117.1.2.6	DisplayString	If applicable to the alert, the database that triggered the alert
Table	.1.3.6.1.4.1.24117.1.2.7	DisplayString	If applicable to the alert, the table that triggered the alert
Timestamp	.1.3.6.1.4.1.24117.1.2.8	DisplayString	The date and time that the alert was detected
Value	.1.3.6.1.4.1.24117.1.2.9	DisplayString	The value detected for the specific metric
Description	.1.3.6.1.4.1.24117.1.2.10	DisplayString	Description of metric
Comments	.1.3.6.1.4.1.24117.1.2.11	DisplayString	User specified comments pertaining to the metric

Create alert response bundles

SQL Diagnostic Manager allows you to create alert response bundles for related alerts. The alert response bundle contains individual alert information for several alerts, summarized and delivered through only one message.

To create an alert response bundle:

1. Click  >**Alert Actions and Responses** in the menu bar.
2. Click **Add** on the Alert Responses tab of the Alert Actions and Responses dialog box.
3. Enter a **Name** for the rule.
4. Check the **Where all the metrics are in the specified list box** condition.
5. Enable the **Send an email to recipient** action.
6. Click **metric** in the Edit the rule description field, and select related metrics to include in the alert response trigger.
7. Click the hyperlink text to select specific conditions for the rule.
8. Click **OK** to finish.

Alerts compatible with the alert response bundles feature

Jobs

- SQL Server Agent Long Running Job (Minutes)
- SQL Server Agent Long Running Job (Percent)
- SQL Server Agent Job Completion
- SQL Server Agent Job Failure
- SQL Server Agent Log

Blocking and Deadlocks

- Blocked Sessions
- Blocked Sessions Wait Time (Seconds)
- Deadlock

Size Databases

- Data File Autogrow
- Database Full (Percent)
- Database Full (Size)

Size Disks

- OS Average Disk Queue Length (Count)
- OS Average Disk Queue Length Per Disk (Count)
- OS Disk Free Space (Size)
- OS Disk Full (Percent)
- OS Disk Time (Percent)
- OS Disk Time Per Disk (Percent)

Error Log

- SQL Server Error Log

Availability Group Databases


- Availability Group Estimated Data Loss (Seconds)
- Availability Group Estimated Recovery time (Seconds)

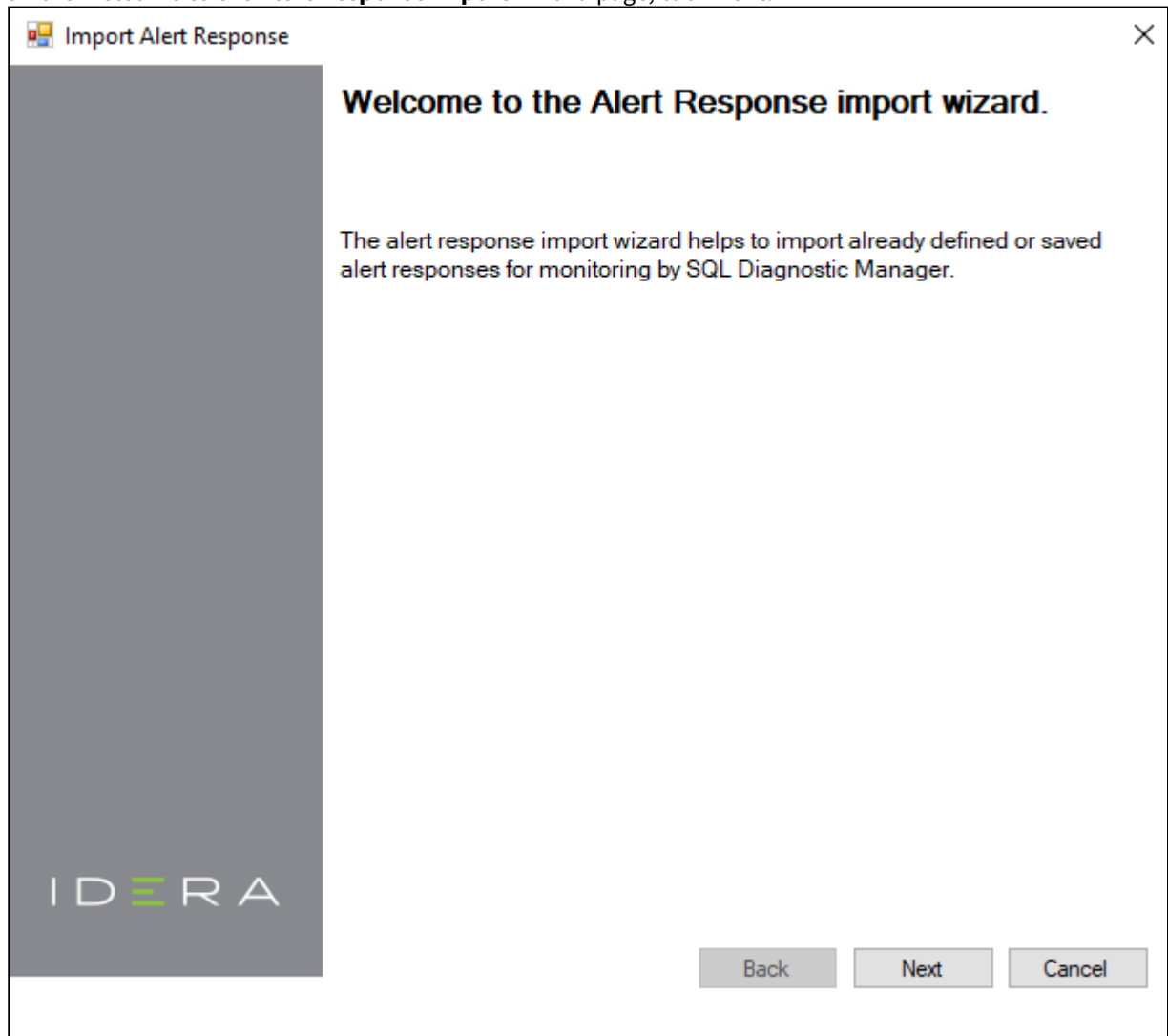
- Availability Group Log Send Queue Size (KB)
- Availability Group Preferred Replica
- Availability Group Redo Queue Size (KB)
- Availability Group Redo Rate (KB/sec)
- Availability Group Role Change
- Availability Group Synchronization Health
- Availability Group Synchronization Performance (Seconds)

Import an alert response

The Alert Response import wizard allows you to import existing alert response actions.


To import an alert response:

1. Open the Alert Actions and Responses window by selecting  > **Alert Actions and Responses** from the SQL Diagnostic Manager console menu and clicking **Import**.
2. On the **Welcome to the Alert Response import wizard** page, click **Next**.



- ✔ You can also access this feature by clicking **Administration>Import/Export** from the navigation pane and then selecting **Alert Response** in the import column.

3. On the **Select Alert Responses to import** page, browse for notification rules' XML files to import.
4. On the **Summary of imported Alert Response** page, you can verify that the selected alert response files were imported correctly. If there are any errors you can view details by clicking **View Validation Errors**.
5. If the imported alert response is valid, click **Finish** to complete the import process and exit the Alert Response import wizard.


 **Export an alert response**

For instructions on exporting an alert response, see [Export an alert response](#).

Export an alert response

SQL Diagnostic Manager allows you to export existing alert response actions.

To export an alert response:

1. Open the Alert Actions and responses window by selecting  > **Alert Actions and Responses** from the SQL Diagnostic Manager console menu. Select the notification rules that you want to export and click **Export**.

- ✔ To export all your alert responses at once, click **Administration > Import/Export** from the navigation pane and select **Alert Response** in the export column.

2. In the **Browse for Folder** dialog box, indicate in which folder you want to save the alert response XML files.
3. Click **OK** in the "Alert response exported to selected directory" notification.

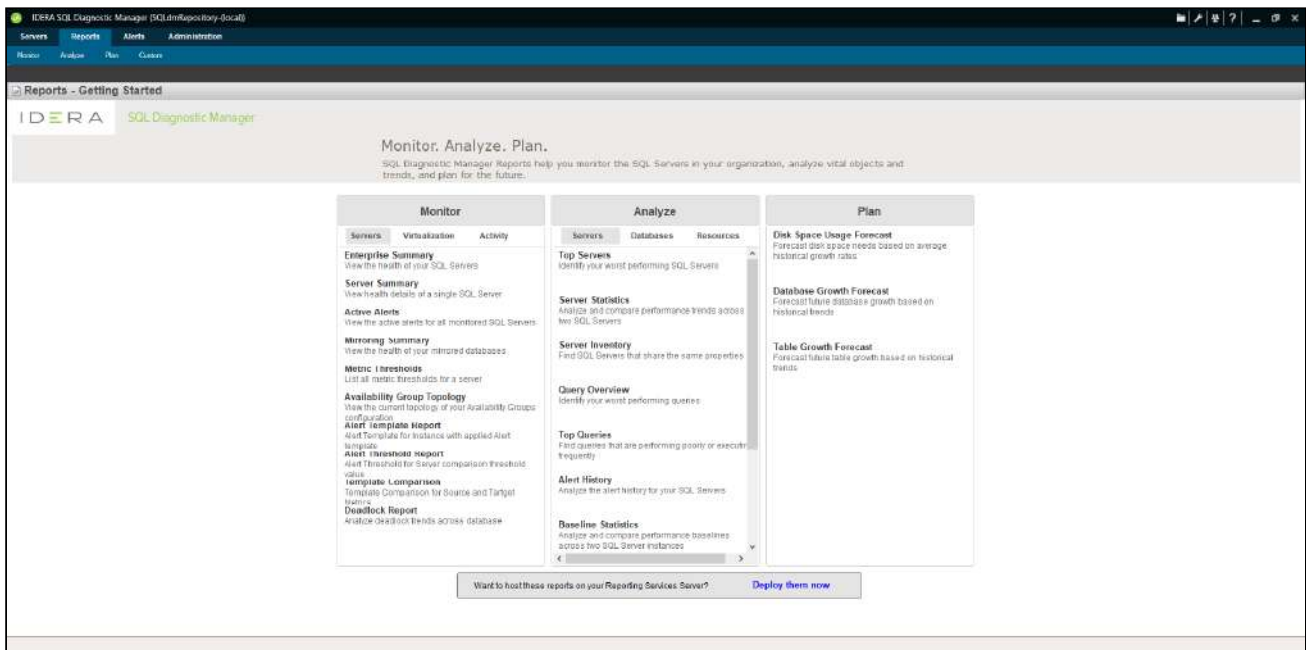
- ✔ **Import an alert response**

For instructions on importing an alert response, see [Import an alert response](#).

Report on SQL Server Performance

The Reports view lets you to create reports that allow you to analyze current and historical performance and statistical data. SQL Diagnostic Manager provides three types of comprehensive reports: [Monitor](#), [Analyze](#), [Plan](#), and [Custom](#).

✔ Reports look best when printed in letter-size format.



Customize reports

After you run a report, you can use the **Report Toolbar** to select the page setup, page width, and print layout to customize your report.

Click **Show Filters** to open the Filter area of the Reports view. Each report has several [filter options](#), such as which SQL Server instances to display, the period of time to report on, and the time intervals to show in the charts.

Deploy reports to Microsoft Reporting Server

You can use the [SQLDM Reports Deployment wizard](#) to deploy specified reports to the Microsoft Reporting Server.

Report filters

Every report in SQL Diagnostic Manager contains filters to allow you to select the information that you want included in your reports. If you want to return to the default filters after making some choices, click **Reset Filters** at the top of the report. Use the **Show Filters/Hide Filters** toggle depending on your preference.

The following list includes all of the filters available, though each report contains different options.

Case Insensitive

Check whether you want SQL Diagnostic Manager to ignore query case for the report results.

Chart Type

Select the type of chart you want to convey your report results.

Compare Server

Optional. Select a SQL Server instance and period of time to compare statistics with your main selected Server and Period.

Compare with same server

Check whether you want to compare the same monitored SQL Server instance using a different metric or period.

Database, Database Name

Select or type the name of one or more databases on which you want to report.

Forecast Type, Forecast Units

Select the type of forecast and number of units to use for projecting future growth trends based on your expected growth rate and sample size.

Drive Name

Select the drive within the monitored SQL Server instance on which you want to report.

End Hour

Select or type the end time of the range from which you want to report trends. SQL Diagnostic Manager includes trends through to the most recent collection time.

Include Batches/SQL Statements/Stored Procedures/System Databases

Check whether you want to include batches, SQL statements, stored procedures, or system databases in your report results.

Metric

Select the metric you want to include in your report.

Min. Duration

Select or type the minimum duration of the queries to display in your report.

Min. CPU/Executions/Reads/Writes

Select or type the minimum number or size of CPU, executions, reads, or writes for the queries you want to display in your report.

Min. Bytes Read/Bytes Written/Data Size/Growth/Transactions

Select or type the minimum number or size of bytes read or written per second, data size, growth, or transactions for the databases you want to display in your report.

Number of Applications

Select or type the number of database applications that you want included in your report.

Number of Databases/Servers

Select or type the number of databases or monitored SQL Server instances you want included in your report.

Number of Queries

Select or type the number of queries that you want returned in the report results.

Action

Select an action type on which you want to report.

Repository User

Select a repository user by whom you want to filter your report.

Workstation

Select a workstation on which you want to report.

Workstation User

Select a workstation user account used to access the workstation on which you want to report.

Period

Select the time range on which you want to report. Choose Custom Range to select the sample months, days, hours, or minutes on which you want to report. The sample choices depend on the type of report selected.

Range

Displays the period of time selected for your report.

Sample

Select the period of time between data points on the graphs that appear in your report. This selection also represents the most current information as of that particular point.

Search Item and Search Value

Select up to four different items and their values to search for in your server inventory, and then type the value you want to search to find. Use a percent (%) character as a wildcard when specifying a Search Value.

Server

Select the monitored SQL Server instance or instances on which you want to report.

Show Problems Only

Check whether you want to include only problems in your report results.

Signature Mode

Check whether you want to group similar queries in your report results.

Start Time, Start Hour

Select or type the start time for the range from which you want to report trends. SQL Diagnostic Manager includes trends through to the most recent collection time.

Tag

Select a sample group for the report based on the [tags](#) that you assigned to those monitored SQL Server instances.

Monitor reports

Monitor reports give you overview information for your monitored SQL Server instances and the virtual machines on which the monitored SQL Server instances are running. In addition, oversee overall activity in your SQL Diagnostic Manager environment through the Activity Monitor Reports.

Access the Reports view

SQL Diagnostic Manager provides two paths to access the Reports view. The first access path is by clicking **Reports** in the Navigation pane. The second access path is by clicking **Go > Reports** . The second path is the only option if you hide your Navigation pane in the SQLDM Console.

Server Monitor Reports

Enterprise Summary

Use the [Enterprise Summary](#) report to view the health of your SQL Servers.

Server Summary

Use the [Server Summary](#) report to view the health of a single SQL Server.

Active Alerts

Use the [Active Alerts](#) report to identify the currently active alerts for all monitored SQL Servers.

Mirroring Summary

Use the [Mirroring Summary](#) report to view the health of your mirrored databases.

Metric Thresholds

Use the [Metric Thresholds](#) report to view all metric thresholds for a SQL Server instance.

Availability Group Topology

Use the [Availability Group Topology](#) report to view the current topology of an AlwaysOn Availability Groups configuration.

Deadlock Report

The [Deadlock Report](#) lets you analyze deadlocks on your SQL Server instance. By analyzing you can better determine where deadlocks are occurring and what changes can have the greatest performance impact.

Alert Template Report

The [Alert Template Report](#) displays the list of instances with Alert Template assigned per instance.

Alert Threshold Report

The [Alert Threshold Report](#) displays all the metrics thresholds of the selected server.

Template Comparison

The [Template Comparison](#) report displays all the metrics thresholds differences between the Source and the Target Template.

Virtualization Monitor Reports

Virtualization Summary

Use the [Virtualization Summary](#) report to view a summary of your virtualized environment.

Virtualization Statistics

Use the [Virtualization Statistics](#) report to view an analysis of your virtualization performance trends for a virtualized machine.

Activity Monitor Reports

Change Log Summary

Use the [Change Log Summary](#) report to view a list of all actions and configuration changes performed in your SQL Diagnostic Manager environment over a specified period of time.

Server monitor reports

Server monitor reports provide an overview of your monitored SQL Server instances.

Server Monitor Reports

Enterprise Summary

Use the [Enterprise Summary](#) report to view the health of your SQL Servers.

Server Summary

Use the [Server Summary](#) report to view the health of a single SQL Server.

Active Alerts

Use the [Active Alerts](#) report to identify the currently active alerts for all monitored SQL Servers.

Mirroring Summary

Use the [Mirroring Summary](#) report to view the health of your mirrored databases.

Metric Thresholds

Use the [Metric Thresholds](#) report to view all metric thresholds for a SQL Server instance.

Availability Group Topology

Use the [Availability Group Topology](#) report to view the current topology of an AlwaysOn Availability Groups configuration.

Deadlock Report

The [Deadlock Report](#) lets you analyze deadlocks on your SQL Server instance. By analyzing you can better determine where deadlocks are occurring and what changes can have the greatest performance impact.

Alert Template Report

The [Alert Template Report](#) displays the list of instances with Alert Template assigned per instance.

Alert Threshold Report

The [Alert Threshold Report](#) displays all the metrics thresholds of the selected server.

Template Comparison

The [Template Comparison Report](#) displays all the metrics thresholds differences between the Source and the Target Template.

Enterprise Summary

The Enterprise Summary report provides an overview of the health and performance of all SQL Server instances monitored by SQL Diagnostic Manager. Click the appropriate instance and SQL Diagnostic Manager displays the Server Summary report.

When to run this report

You can run the Enterprise Summary report to view a quick overview of the health and performance of all monitored instances. This report allows you to see the status of all your SQL Server instances at a glance.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Enterprise Summary report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Server Summary

The Server Summary report provides a detailed overview of the health and performance of a specific SQL Server, such as SQL Server status, and File Use. This report displays graphs of the following metrics: **Response Time**, **SQL Server CPU Activity**, **OS Memory Usage**, and **Disk Busy**. Also, it provides the databases information of the selected server: **Database Name**, **Creation Date**, **Data + Log Size (MB)**, and **Status**.

When to run this report

You should run the Server Summary report to view the detailed performance and health information of a single SQL Server.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Server Summary report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Active Alerts

The Active Alerts report lists all outstanding, active alerts that SQL Diagnostic Manager has raised against your SQL Server instances. Select a Server or a Tag from the drop-down lists to run the Active Alerts Report. This report lists all the Alerts, the Status, the Description, and the Details of all active alerts.

When to run this report

You should run the Active Alerts report to view all alerts currently active on your monitored instances. You can sort the alerts by severity level, so you can focus on the most critical alerts first.

How SQL Diagnostic Manager calculates metric values on this report

The values on the Active Alerts report correlate with the related SQL Server metric.

Mirroring Summary

The Mirroring Summary report allows you to view the current mirroring sessions across all or selected tags or servers.

The data for this report comes from SQL Server system views for database mirroring. The SQL Diagnostic Manager alert thresholds are used to highlight existing issues as defined by your configured alert settings.

When to run this report

You should run the Mirroring Summary report to view a quick overview of the health of your mirrored databases.

How SQL Diagnostic Manager calculates metric values on this report

The values on the Mirroring Summary report correlate with the related SQL Server metric.

Metric Thresholds

The Metric Thresholds report provides a list of metric thresholds for a monitored SQL Server instance. To successfully monitor OS metrics, enable [WMI or OLE automation](#) on the monitored instance.

In the Metric Thresholds report results, the table displays the Metric name, the Threshold Instance, and the threshold values (Informational, Warning, and Critical).

When to run this report

You should run the Metric Thresholds report to review your current Informational, Warning, and Critical threshold settings for the selected SQL Server instance. This report also lists whether a metric is enabled, helping you decide what action is necessary for you to get the most out of SQL Diagnostic Manager.

How SQL Diagnostic Manager calculates metric values on this report

The values on the Metric Thresholds report correlate with the related SQL Server metric.

Availability Group Topology

The Availability Group Topology report allows users to view the current topology of an AlwaysOn availability group's configuration. The report displays all of the availability groups on the server, replicas that participate in the group, and the databases within the group.

You can filter results by availability group only. The report lists detailed information for availability replicas and corresponding databases.

Availability replica table

Provides information on availability Replica, Roles assigned, Failover mode, Availability Modes selected, and the Secondary Connection Mode.

Availability databases table

Provides information on the Database, Failover readiness, and Synchronization State of the availability databases contained within the selected availability replica.

When to run this report

You should run the Availability Group Topology report to keep track of modifications made to your current AlwaysOn availability groups and avoid any potential setbacks to your high-availability and disaster recovery strategies.

Deadlock Report

The Deadlock report lets you analyze deadlocks on your SQL Server instance. The report filters the data by Instance Name, followed by **Application Name**, **Database Name**, **SQL Statement**, **Collection Date**, **XDL Data**, **Deadlock ID**, **Login Name**, and **Host Name**.

This report lists all the deadlock occurrences historically and in detail.

When to run this report

By analyzing the deadlocks, you can better determine where deadlocks are occurring and analyze the changes can have the greatest performance impact.

Alert Template Report

The Alert Template report lists the last applied template in the selected servers, or selected Tag.

To run this report simply, select a tag, or a server from the drop-down list, and click **Run Report**.

Alert Threshold Report

The Alert Threshold Report lists which thresholds have changed from the last applied template.

To run this report simply, select a server from the drop-down list, and click **Run Report**.

Template Comparison

The template comparison report displays all the metric thresholds differences for a selected Source and a Target template.

To run this report simply, select a Source Template from the drop-down list, chose a Target Template, and click **Run Report**.

Virtualization monitor reports

Virtualization monitor reports provide overview information for the virtual machines on which your monitored SQL Server instances are running. Reports are compatible for Hyper-V or VMWare VMs. Click the links below for more information.

Virtualization Monitor Reports

Virtualization Summary

Use the [Virtualization Summary](#) report to view a summary of your virtualized environment.

Virtualization Statistics

Use the [Virtualization Statistics](#) report to view an analysis of your virtualization performance trends for a virtualized machine.

Virtualization Summary

The Virtualization Summary report provides an overview of the health and performance of the virtualized environment for the selected SQL Server instance. This report includes information about the VM as well as configuration of the host server. Click the appropriate instance and SQL Diagnostic Manager displays the Virtualization Summary report.

When to run this report

You can run the Virtualization Summary report to view a quick overview of the health and performance of the virtualized environment for your selected monitored instance. Run this report when you want a quick glance at the VM information for this instance.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Virtualization Summary report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)

Virtualization Statistics

The Virtualization Statistics report provides an analysis of your virtualization performance trends for a virtualized machine monitored by SQL Diagnostic Manager. Select the appropriate instance and SQL Diagnostic Manager displays the Virtualization Statistics report using your selected chart type.

When to run this report

You can run the Virtualization Statistics report when you are concerned about a virtualized machine and its performance. This report allows you to see the CPU, memory, disk, and network performance metrics for the selected VM and period of time.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Virtualization Statistics report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)

Activity monitor reports

Activity monitor reports provide overview information on all actions and configuration changes performed in SQL Diagnostic Manager. Click the link below for more information.

Activity Monitor Reports

Change Log Summary

Use the [Change Log Summary](#) report to view a summary of key actions performed in your SQL Diagnostic Manager environment.

Change Log Summary

The Change Log Summary report lets you view a list of all actions and configuration changes performed in your SQL Diagnostic Manager environment over a specified period of time and through various [filter options](#). The report contains detailed information, Date, Action, Object Name, Workstation, Repository User, and Change Summary. For Example, users are able to view all changes performed by a specific repository user, by expanding Details.

When to run this report

You should run the Change Log Summary report to audit overall activity in your SQL Diagnostic Manager environment. This report allows you to analyze key changes that might hinder your performance or represent a risk.

Analysis reports

Analysis reports provide information that allows you to analyze the current effectiveness of your SQL Servers. Click the links below for more information.

Access the Reports view

SQL Diagnostic Manager provides two paths to access the Reports view. The first access path is by clicking **Reports** in the Navigation pane. The second access path is by clicking **Go > Reports**. The second path is the only option if you hide your Navigation pane in the SQL Diagnostic Manager Console.

Server Analysis Reports

Top Servers

Use the [Top Servers](#) report to identify your worst performing SQL Servers.

Server Uptime

Use the [Server Uptime](#) report to review the uptime and downtime of your monitored SQL Server.

Server Statistics

Use the [Server Statistics](#) report to analyze and compare performance trends across two SQL Servers.

Server Inventory

Use the [Server Inventory](#) report to find SQL Servers that share the same properties.

Query Overview

Use the [Query Overview](#) report to identify your worst performing queries.

Top Queries

Use the [Top Queries](#) report to find queries that are performing poorly or executing too frequently.

Alert History

Use the [Alert History](#) report to analyze the alert history for a SQL Server.

Baseline Statistics

Use the [Baseline Statistics](#) report to review and compare metric and baseline values across two SQL Server instances.

Query Wait Statistics

Use the [Query Wait Statistics](#) report to analyze different wait type categories on your SQL Server instance.

Database Analysis Reports

Top Databases

Use the [Top Databases](#) report to identify your worst performing databases.

Database Statistics

Use the [Database Statistics](#) report to analyze and compare performance trends across two databases.

Top Database Applications

Use the [Top Database Applications](#) report to find database applications that consume system resources.

Mirroring History

Use the [Mirroring History](#) report to analyze the event history for a mirrored database.

Top Tables by Growth

Use the [Top Tables by Growth](#) report to identify the fastest growing tables.

Top Tables by Fragmentation

Use the [Top Tables by Fragmentation](#) report to identify the most fragmented tables.

Tempdb Statistics

Use the [Tempdb Statistics](#) report to analyze performance statistics for your tempdb database.

Transaction Log Statistics

Use the [Transaction Log Statistics](#) report to analyze the use of your log files.

Availability Group Statistics

Use the [Availability Group Statistics](#) report to view the historical health of your availability groups, availability replicas, and availability databases.

Resource Analysis Reports

Session Statistics

Use the [Session Statistics](#) report to track key session and network performance metrics over time.

Detailed Session Report

The [Detailed Session](#) report provides detailed session information on your selected SQL Server instance.

CPU Statistics

Use the [CPU Statistics](#) report to track key CPU performance metrics.

Disk Details

Use the [Disk Details](#) report to track key disk metrics.

Disk Space Usage

Use the [Disk Space Usage](#) report to obtain an overview of the available disk space across your SQL Server environment.

Disk Space History

Use the [Disk Space History](#) report to view the history of activity for a specific disk.

Disk Statistics

Use the [Disk Statistics](#) report to track key disk performance metrics.

Replication Statistics

Use the [Replication Statistics](#) report to track key replication performance metrics.

Memory Statistics

Use the [Memory Statistics](#) report to track key memory performance metrics.

Server analysis reports

Server analysis reports provide information that allows you to analyze the current effectiveness of your SQL Servers. Click the links below for more information.

Server Analysis Reports

Top Servers

Use the [Top Servers](#) report to identify your worst performing SQL Servers.

Server Uptime

Use the [Server Uptime](#) report to review the uptime and downtime of your monitored SQL Server.

Server Statistics

Use the [Server Statistics](#) report to analyze and compare performance trends across two SQL Servers.

Server Inventory

Use the [Server Inventory](#) report to find SQL Servers that share the same properties.

Query Overview

Use the [Query Overview](#) report to identify your worst performing queries.

Top Queries

Use the [Top Queries](#) report to find queries that are performing poorly or executing too frequently.

Alert History

Use the [Alert History](#) report to analyze the alert history for a SQL Server.

Baseline Statistics

Use the [Baseline Statistics](#) report to review and compare metric and baseline values across two SQL Server instances.


Query Wait Statistics

Use the [Query Wait Statistics](#) report to analyze the different query waits in a SQL Server.

Top Servers

The Top Servers report identifies your worst performing SQL Server instances based on the number of active alerts for the current monitored instances, response time, CPU usage, memory usage, and disk usage.

You can further filter your monitored instances by wait time with waits that exceed a defined **Wait Threshold** in milliseconds. This report displays the values of the following metrics: **Top Servers by Active Alerts**, **Top Servers by Response Time**, **Top Servers by CPU Usage**, **Top Servers by Memory Usage**, **Top Servers by Disk Usage**, and **Top Servers by Wait**.

 A value of N/A indicates that the Collection service failed to collect data due to disabled OS metrics or a non-responsive WMI server. You must [enable OS metrics monitoring](#) to receive results.

When to run this report

Run the Top Servers report daily to keep ahead of critical issues that can cause an unavailable instance. Use this report to compile a list of instances that need immediate attention.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Top Servers report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric. You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237

Server Statistics

The Server Statistics report lets you analyze and compare performance trends across two SQL Server instances.

You can select a Chart Type from the following drop-down list to be displayed in the report: **Response Time**, **% CPU Usage**, **Memory Usage (GB)**, and **% Disk Busy**. The Master Metrics table shows the comparison of the selected instances for all the mentioned metrics over a specified period of time.

When to run this report

You should run the Server Statistics report when you want to trend key performance metrics over time for one or two specific instances, in response to a known problem or a routine health check. For example, if the same third party application uses two different databases but only one database has reported issues, then comparing their key performance metrics can help you diagnose the issue.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Server Statistics report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237

Server Inventory

The Server Inventory report lets you compile a list of SQL Server instances that share up to seven different properties: **Server Name**, **Server Version**, **OS Version**, **Number of Processors**, **Physical Memory**, and **Clustered**.

✔ You can use % as a wildcard in the **Search Value** field.

When to run this report

Run the Server Inventory report when you need to identify which monitored instances require software or hardware upgrades, or simply want to know how many instances of a SQL Server version are deployed.

Deleted SQL Server instances

When you choose to delete a SQL Server instance, SQL Diagnostic Manager displays the confirmation message, "Would you like to retain the data collected for the selected instance?" When you click **Yes**, SQL Diagnostic Manager retains data for the instance and includes it in some reports, such as the Server Inventory report.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Server Inventory report correlate with the related SQL Server metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)

Query Overview

The Query Overview report identifies the worst performing queries run on a specific SQL Server instance over time, including the call volume of each query. This report requires that you enabled Query Monitoring in the SQL Diagnostic Manager Console and have collected sufficient query data for the specified time period. This report displays the values of the Stored Procedures, SQL Statements, Batches, and the Query Overview.

When to run this report

You should run the Query Overview report when you want to determine whether SQL Server queries run through Microsoft SQL Server tools or third-party applications are impacting the performance of your monitored instances. This report can help you further diagnose a known performance issue or proactively identify queries whose resource consumption may cause issues in the future.

How SQL Diagnostic Manager calculates metric values on this report

The values on the Query Overview report correlate with the related SQL Server metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)

Top Queries

The Top Queries report lets you compile a list of queries based on call frequency, duration of execution, CPU usage, and the number of reads and writes performed on the databases hosted by the specified SQL Server instance. You can define minimum thresholds for each of these performance metrics and then see which queries match or exceed the selected values. This report requires that you enabled Query Monitoring in the SQL Diagnostic Manager Console and have collected sufficient query data for the specified time period.

When viewing results, note that you can click the SQL statement to view the [Query Statistics report](#), which provides details on an individual SQL statement or signature as collected by the Query Monitor.

This report displays the Statements by **Top Queries By Frequency**, **Top Query by Avg. Duration**, **Top Queries By CPU**, **Top Queries By Reads**, and **Top Queries By Writes**.

When to run this report

You should run the Top Queries report on a routine basis, such as every week, to proactively identify potential performance issues that may be caused by queries. You can also use this report to track historic performance trends and determine the relationship between query usage and your database performance.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Top Queries report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237

Query Statistics

The Query Statistics report lets you view details for an individual SQL statement or signature as collected by the Query Monitor. This report requires that you enabled Query Monitoring in the SQL Diagnostic Manager Console and have collected sufficient query data for the specified time period.

When to run this report

You should run the Query Statistics report to analyze an individual SQL statement or signature for a monitored instance.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Query Statistics report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)
- [What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237](#)

Alert History

The Alert History report lets you review the alerts that SQL Diagnostic Manager has raised for the selected SQL Server instance over a specified period of time. You can also review alerts, the time, status and the description of each one of them.

When to run this report

You should run the Alert History report when you want to analyze the historical health of the SQL Server by identifying and tracking critical values for key performance metrics. Each raised alert represents a point in time when a metric reached or exceeded the specified threshold.

By default, SQL Diagnostic Manager alerts on many performance-related SQL Server metrics. You can also [configure alerts](#) and [create custom counters](#) to track specific performance data.

How SQL Diagnostic Manager calculates metric values on this report

The values on the Alert History report correlate with the related SQL Server metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)

Baseline Statistics

The Baseline Statistics report lets you analyze and compare baselines within a single SQL Server instance and across two instances. When viewing baseline statistics for a monitored SQL Server instance, you can compare the baseline metric values at two different times or two different metrics at the same time. Include another instance and you can compare baselines values occurring at the same time or different times.

When to run this report

You should run the Baseline Statistics report when you want to view trends in the average value of a metric for a SQL Server instance and how this value changes over time to help you in capacity planning. This report also provides insight into the utilization of your different instances by comparing the baselines over time.

Report selections change based on the selected Period

The Baseline Statistics report **Period** indicates the date range for which you want to see results. If you select **Today** or **Custom Range**, SQL Diagnostic Manager displays additional fields that allow you to filter results by the start and end hours.

✔ If you select **Custom Range** as the **Period**, and the start and end dates are both for the same day, SQL Diagnostic Manager calculates the selection the same as if you select **Today** as the **Period**.

How SQL Diagnostic Manager calculates metric values on this report

The baseline values are the most recent calculations for the selected period. Note that the **Metric Value** is an average of all data for the selected metric. For example, if you select **Hours** from the **Sample** drop-down list, the Baselines Statistics report grid includes 24 rows for a single day, one row for each hour from 00:00 to 23:00. Each row displays a metric value which is the average of the selected metric for that hour, such as 05:00 to 05:59:59, and the baseline values which are calculated for the entire 24-hour day. As a result, SQL Diagnostic Manager displays 24 rows of different metric values but equal baseline values.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)

Query Wait Statistics

The Query Wait Statistics report lets you analyze different wait type categories on your SQL Server instance. By analyzing these waits you can better determine where your biggest bottlenecks are occurring and what changes could have the greatest performance impact. You can select from various wait type categories such as **Backup**, **Excluded**, **I/O**, **Lock**, **Memory**, **Non-I/O Page Latch**, **Non-Page Latch**, **Transaction Log**, and **Other**.

This report requires that you enabled Query Monitoring in the SQL Diagnostic Manager Console and have collected sufficient query data for the specified time period.

When to run this report

You should run the Query Wait Statistics report on a routine basis, such as every week, to proactively identify potential performance issues that may be caused by queries.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Query Wait Statistics report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.


- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237

Server Uptime

The Server Uptime report lets you review the uptime and the downtime of a server that SQL Diagnostic Manager raises for the selected SQL Server instance over a specified period of time.

To run the Server Uptime report, follow these steps:

1. Select the instance you want to review from the Server drop-down menu.
2. Specified a Period of time (Today, Last 7 Days, Last 30 Days, Last 365 Days, and Custom Range).

 When you select Custom Range, the Select Custom Range windows open to specify the Period of Time of your preference. The date range is limited for which data was collected for your monitored servers.

3. Click **Run Report**.

Database analysis reports

Database analysis reports provide information on databases that allows you to analyze the current effectiveness of your SQL Servers. Click the links below for more information.

Database Analysis Reports

Top Databases

Use the [Top Databases](#) report to identify your worst performing databases.

Database Statistics

Use the [Database Statistics](#) report to analyze and compare performance trends across two databases.

Top Database Applications

Use the [Top Database Applications](#) report to find database applications that consume system resources.

Mirroring History

Use the [Mirroring History](#) report to analyze the event history for a mirrored database.

Top Tables by Growth

Use the [Top Tables by Growth](#) report to identify the fastest growing tables.

Top Tables by Fragmentation

Use the [Top Tables by Fragmentation](#) report to identify the most fragmented tables.

Tempdb Statistics

Use the [Tempdb Statistics](#) report to analyze performance statistics for your tempdb database.

Transaction Log Statistics

Use the [Transaction Log Statistics](#) report to analyze the use of your log files.

Availability Group Statistics

Use the [Availability Group Statistics](#) report to view the historical health of availability groups, availability replicas, and availability databases.

Top Databases

The Top Databases report identifies your worst performing SQL Server databases based on the size of the database, the growth rate of the database, and the number of reads, writes, and transactions per second performed on the database.

You can further filter your monitored databases by wait time with waits that exceed a defined **Wait Threshold** in milliseconds. Use this report to compile a list of databases that used most often or have the heaviest loads, these databases are shown in different tables: **Top Databases By Data Size**, **Top Databases By Data Growth**, **Top Databases By Reads Per Second**, **Top Databases By Writes Per Second**, and **Top Databases By Transactions Per Second**.

 You can use % as a wildcard in the **Search Value** field.

When to run this report

You should run the Top Databases report on a routine basis, such as once a week. This report helps you:

- Plan for upcoming maintenance
- Identify issues that may impact third-party application performance even though the host SQL Server instance remains healthy
- Compare database performance across your enterprise

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Top Databases report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237

Database Statistics

The Database Statistics report lets you analyze and compare performance trends, such as the percentage of growth, across two SQL Server databases. You can also compare performance metrics at different points in time for the same database. You can select a metric from the drop-down Chart Type list to be displayed in a chart on your report: **Data File Size MB**, **Log File Size MB**, **Data Growth %**, **Reads Per Second**, **Writes Per Second**, and **Transactions Per Second**.

When to run this report

You should run the Database Statistics report when attempting to understand performance differences or troubleshooting database issues.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Database Statistics report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237

Top Database Applications

The Top Database Applications report lets you find the applications that are consuming the highest amount of your system resources, this report displays the value for **Top Applications By CPU**, **Top Applications By Reads**, and **Top Applications By Writes**. You can define minimum thresholds for each of these performance metrics and then see which applications match or exceed the selected values. This report requires that you enabled Query Monitoring in the SQL Diagnostic Manager Console and have collected sufficient query data for the specified time period.

✔ You can use % as a wildcard in the **Search Value** field.

When to run this report

You should run the Top Database Applications report on a routine basis, such as every week, to proactively identify potential performance issues that may be caused by third-party applications. You can also use this report to track historical performance trends and determine the relationship between third-party application usage and your database performance.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Top Database Applications report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How is the Worst Performing and Most Frequent Queries information gathered?](#) - Solution #00000412
- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers?](#) - Solution #00000055

Mirroring History

The Mirroring History report lets you view the status of select mirroring metrics at the time of each SQL Diagnostic Manager scheduled refresh for the specified period.

You can filter the results by Time, Tag, Server, Database, and also check to view only problems as defined in your alerts. Fields showing metrics corresponding to alerts are color-coded according to the alert.

When to run this report

You should run the Mirroring History report when you want to analyze the historical health of your mirrored SQL Server databases by identifying and tracking critical values for key performance metrics. Each red or yellow value in the report grid corresponds to a raised alert. These visual warnings represent points in time when the metric reached or exceeded the specified alert threshold.

By default, SQL Diagnostic Manager alerts on many performance-related SQL Server metrics. You can also [configure alerts](#) and [create custom counters](#) to track specific performance data.

Transaction Log Statistics

The Transaction Log Statistics report provides data about the database transaction log size and activity on a per-database basis. You can display your results in hours or days, depending on the sample range selected. Note that SQL Diagnostic Manager displays the Cache Reads and Flushes chart types as data per second.

Select a metric from the Chart Type drop-down list to be displayed in this report: **Cache Hit Ratio, Cache Reads, File Size, Flush Waits, Flushes, and Kilobytes Flushed.**

When to run this report

You should run the Transaction Log Statistics report when attempting to troubleshoot performance problems related to transaction logs.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Transaction Log Statistics report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237

Top Tables by Growth

The Top Tables by Growth report identifies the fastest growing tables in a database. You can select a specific database to analyze or evaluate all databases on a specific SQL Server instance. You can also choose a growth metric and its minimum threshold to see which tables match or exceed the selected value. This report displays the fastest growing tables **by Size Growth %**, **by Row Growth%**, **by Number of Rows**, **by Total Size**, **by Data Size**, **by Text Size**, and **by Index Size**.

When to run this report

You should run the Top Tables by Growth report on a routine basis, depending on how heavily your company uses the host SQL Server instance. Use this report to see how rapidly (or slowly) tables are growing over time. ***If database capacity on a given instance becomes limited***, run this report to see a list of tables that are the top contributors and decide whether you want to move them to another database.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Top Tables by Growth report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237

Top Tables by Fragmentation

The Top Tables by Fragmentation report identifies the most fragmented tables in a database, select a specific database to analyze or evaluate all databases on a specific SQL Server instance. You can also choose a fragmentation metric and its minimum threshold to see which tables match or exceed the selected value. Also, this report display the **Top Tables by Logical Fragmentation** and **Top Tables by Data Size**.

When to run this report

You should run the Top Tables by Fragmentation report on a routine basis, depending on how mission-critical or sensitive the data is and how heavily your company uses the host database. Use this report to identify the tables that you want to optimize for improved performance.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Top Tables by Fragmentation report correlate with the related SQL Server metric.


However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412](#)
- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)

Tempdb Statistics

The Tempdb Statistics report lets you view space utilization and data throughput statistics over time for your Tempdb database. You can focus on a single chart type, such as data file growth, and track its value increase over a specified period of time. This report allows you to obtain statistics through different metrics or chart types: **Data File Size MB, Data File Growth, Reads Per Second, Writes Per Second, Transactions Per Second, User Objects MB, Internal Objects MB, Version Store MB, and Mixed Extents MB.**

 The Tempdb Statistics report is available for users with monitored SQL Server instances on SQL Server 2005 or later.

When to run this report

You should run the Tempdb Statistics report to view different space utilization and data throughput statistics currently occurring on your monitored SQL Server instances.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Tempdb Statistics report are averages of data from the system DMV `sys.dm_db_file_space_usage` on the monitored SQL Server.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)

Availability Group Statistics

The Availability Group Statistics report allows users to view the health of their availability groups, availability replicas, and availability databases over a specific time range.

You can filter results by tag, availability group, period, sample, server, and chart type. The report compiles crucial health metrics that include the **Redo Rate (KB/sec)**, **Redo Queue Size (KB)**, **Log Send Rate (KB/sec)**, and **Log Send Queue Size (KB)**.

The Availability Group Statistics report also provides DBAs with a graphical representation of a selected metric and a statistics table with weighted averages for relevant metrics.

For individual details on availability databases per replica, click the expandable command button.

- ✔ Access the Availability Group Topology report by clicking a specific group link while viewing the Availability Group Statistics report.

When to run this report

You should run the Availability Group Statistics report when you want to analyze the historical health of your availability groups and corresponding components. This report helps you to ensure the highest level of availability by identifying and tracking critical health metrics.

Resource analysis reports

Resource analysis reports provide information that focuses on key performance metrics of your SQL Servers. Click the links below for more information.

Resource Analysis Reports

Session Statistics

Use the [Session Statistics](#) report to track key session and network performance metrics over time.

Detailed Session Report

The [Detailed Session](#) report provides detailed session information on your selected SQL Server instance.

CPU Statistics

Use the [CPU Statistics](#) report to track key CPU performance metrics.

Disk Details

Use the [Disk Details](#) report to track key disk metrics.

Disk Space Usage

Use the [Disk Space Usage](#) report to view disk space information, you can filter the information by selecting one or more disks of your environment.

Disk Space History

Use the [Disk Space History](#) report to view disk activity history across your environment.

Disk Statistics

Use the [Disk Statistics](#) report to track key disk performance metrics.

Replication Statistics

Use the [Replication Statistics](#) report to track key replication performance metrics.

Memory Statistics

Use the [Memory Statistics](#) report to track key memory performance metrics.

Session Statistics

The Session Statistics report lets you view trends for sessions and network connectivity, such as transactions per minute, on a monitored SQL Server instance over a specified period of time. Check **Show Tabular Data** to display results in a single tabular or grid format and check **Show Baseline** to include baseline data points in the Session Statistics report. For additional information about baseline configuration, see [Configure server baseline options](#).

The Session Statistics report displays the following graphs: **User Processes and Computers**, **Logins per Minute**, **Network Packets per Minute**, **Transactions per Minute**, and **Oldest Open Transaction (minutes)**.

When to run this report

You should run the Session Statistics report to analyze the sessions and network connectivity for a monitored instance. High logins or transactions per minute may indicate a need for increased resources.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Session Statistics report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Detailed Session Report

The Detailed Session report provides detailed session information of your SQL Server instance. This report trigger the list of **Session IDs**, **User Name**, **Host Name**, **Application**, **Login Time**, and the **Last Activity** time.

When to run this report

You should run the Detailed Session report to analyze the sessions for a monitored instance.

CPU Statistics

The CPU Statistics report lets you view trends for CPU usage on a monitored SQL Server instance over a specified period of time. Check **Show Tabular Data** to display results in a single tabular or grid format and check **Show Baseline** to include baseline data points in the CPU Statistics report. For additional information about baseline configuration, see [Configure server baseline options](#).

The CPU Statistics reports displays the following graphs: **CPU Usage (%)**, **Processor Queue Length**, **SQL Compilations**, **SQL Recompilations**, **Locks Waits per Minute**, and **Table Lock Escalations**,

When to run this report

You should run the CPU Statistics report to analyze contributors to CPU usage for a monitored instance.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the CPU Statistics report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Memory Statistics

The Memory Statistics report lets you view trends for memory usage on a monitored SQL Server instance over a specified period of time. To successfully monitor OS metrics, make sure that you enable [WMI or OLE automation](#) on the monitored instance. Check **Show Tabular Data** to display results in a single tabular or grid format.

Check **Show Baseline** to include baseline data points in the Memory Statistics report. For additional information about baseline configuration, see [Configure server baseline options](#).

When to run this report

You should run the Memory Statistics report to analyze the memory needs of a monitored instance. This can help you identify problems with current usage or to plan for future increases in memory.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Memory Statistics report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Disk Details

The Disk Details report lets you view key disk metrics of a monitored SQL Server instance and specific disk drive over a specified period of time. To successfully monitor OS metrics, make sure that [WMI/OLE automation is enabled](#) on the monitored instance. Check **Show Tabular Data** to display results in a single tabular or grid format.

This report generates the following graphs: **Disk Time Per Read (MS)**, **Disk Time Per Transfer (MS)**, **Disk Time Per Write (MS)**, **Disk Reads Per Second**, **Disk Transfers Per Second**, and **Disk Writes Per Second**.

When to run this report

You should run the Disk Details report to analyze the six key metrics for disk speed, such as disk time per read or disk reads per second.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Disk Details report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)
- [How to customize existing reports used with Reporting Services? - Solution #00000307](#)



Customize your report and use the Chart filter to include either the average or maximum disk details.

Disk Statistics

The Disk Statistics report lets you view trends for the disk activity of a monitored SQL Server instance over a specified period of time. To successfully monitor OS metrics, make sure that you enable [WMI or OLE automation](#) on the monitored instance. Check **Show Tabular Data** to display results in a single tabular or grid format and check **Show Baseline** to include baseline data points in the Disk Statistics report. For additional information about baseline configuration, see [Configure server baseline options](#).

This report generates the following graphs: **IO Activity %**, **Work Files Per Minute**, **Tempdb Size (MB)**, **Page Reads Per Second**, **Page Writes Per Second**, **Page Splits Per Minute**, **Page Lookups Per Minute**, **% Disk Busy**, and **Disk Queue Length**.

When to run this report

You should run the Disk Statistics report to analyze the disk activity such as work files per minute or page read/writes per second.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Disk Statistics report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Replication Statistics

The Replication Statistics report lets you view trends for the replicated transactions on a monitored SQL Server instance over a specified period of time. Check **Show Tabular Data** to display results in a single tabular or grid format and check **Show Baseline** to include baseline data points in the Replication Statistics report. For additional information about baseline configuration, see [Configure server baseline options](#).

This report generates the following graphs: **Replication Statistics** and **Maximum Subscription Latency**.

When to run this report

You should run the Replication Statistics report to analyze the efficiency of replication between databases.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Replication Statistics report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Disk Space Usage

The Disk Space Usage report provides an overview of the available disk space, this information can be filter by selecting one or more disks of your SQL Server environment. To access this report, click **Reports** in the Navigation pane, then in the Analyze menu select **Resources > Disk Space Usage**.

Select one of the following metrics from the Chart Type drop-down menu to be displayed in the report: **SQL Data Used MB**, **SQL Data Free MB**, **SQL Log File MB**, **Non-SQL Disk Usage MB**, **Free Disk Space in MB**, **Disk Reads Per Sec**, and **Disk Writes Per Sec**.

When to run this report

You can run the Disk Space Usage report to understand where you have the most and the least disk space available in your environment.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Disk Space Usage report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Deploy report to Microsoft Reporting Server

You can use the [SQLdm Reports Deployment wizard](#) to deploy the Disk Space Usage report to the Microsoft Reporting Server.

Disk Space History

The Disk Space History report provides the history of activity for a specific disk. To access this report, click **Reports** in the Navigation pane, then select **Resources > Disk Space History** in the Analyze menu.

When to run this report

You can run the Disk Space History report to locate a new database by analyzing available disk space and level of activity.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Disk Space History report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Deploy report to Microsoft Reporting Server

You can use the [SQLdm Reports Deployment wizard](#) to deploy the Disk Space History report to the Microsoft Reporting Server.

Plan reports

Planning reports allow you to plan for future needs based on current trends.

Access the Reports view

SQL Diagnostic Manager provides two paths to access the Reports view. The first access path is by clicking **Reports** in the Navigation pane. The second access path is by clicking **Go > Reports**. The second path is the only option if you hide your Navigation pane in the SQL Diagnostic Manager Console.

Available reports

Disk Space Usage Forecast

Use the [Disk Space Usage Forecast](#) report to forecast disk space needs based on average historical growth rates.

Database Growth Forecast

Use the [Database Growth Forecast](#) report to forecast future database growth based on historical trends.

Table Growth Forecast

Use the [Table Growth Forecast](#) report to forecast future table growth based on historical trends.

Disk Space Usage Forecast

The Disk Space Usage Forecast report lets you predict how much disk space you need in the future based on current and historical growth rate trends. For more accurate forecasting, select the largest possible period of historical data and the shortest possible projection into the future.

- ✔ The Disk Space Usage Forecast report does not display the maximum capacity of the disk. You can use this report to estimate how much space you need, but it does not indicate when existing space runs out.

When to run this report

Run the Disk Space Usage Forecast report to help anticipate your needs and plan for future acquisition of disk space.

Available forecast types

You can choose Linear or Exponential (Aggressive) forecasting types when you run this report.

Linear forecasting follows the trend of the available data and extends the forecast out in a straight line by the number of forecast units.

Exponential forecasting tends to exaggerate the trend that exists in the data. This allows you to more easily see if your data (disk space usage) is trending toward or away from a critical threshold. If the rate of growth is not constant, the exponential forecast type gives a more accurate forecast of future growth.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Disk Space Usage Forecast report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Database Growth Forecast

The Database Growth Forecast report lets you predict how large selected databases may grow based on current and historical growth rate trends. For more accurate forecasting, select the largest possible period of historical data and the shortest possible projection into the future.

When to run this report

You should run the Database Growth Forecast report to help anticipate your needs and plan for future database expansion.

Available forecast types

You can choose Linear or Exponential (Aggressive) forecasting types when you run this report.

Linear forecasting follows the trend of the available data and extends the forecast out in a straight line by the number of forecast units.

Exponential forecasting tends to exaggerate the trend that exists in the data. This allows you to more easily see if your data (database growth) is trending toward or away from a critical threshold. If the rate of growth is not constant, the exponential forecast type gives a more accurate forecast of future growth.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Database Growth Forecast report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412](#)
- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers? - Solution #00000055](#)
- [How does SQL Diagnostic Manager calculate the reorganization percentage for database tables? - Solution #00000301](#)
- [What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237](#)

Table Growth Forecast

The Table Growth Forecast report lets you predict how large selected tables may grow based on current and historical growth rate trends. For more accurate forecasting, select the largest possible period of historical data and the shortest possible projection into the future.

When to run this report

You should run the Table Growth Forecast report to help anticipate your needs and plan for future creation of additional tables.

Available forecast types

You can choose Linear or Exponential (Aggressive) forecasting types when you run this report.

Linear forecasting follows the trend of the available data and extends the forecast out in a straight line by the number of forecast units.

Exponential forecasting tends to exaggerate the trend that exists in the data. This allows you to more easily see if your data (table growth) is trending toward or away from a critical threshold. If the rate of growth is not constant, the exponential forecast type gives a more accurate forecast of future growth.

How SQL Diagnostic Manager calculates metric values on this report

Most values on the Table Growth Forecast report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Support Portal](#) for more information about the algorithm used to calculate a specific metric.

- [How is the Worst Performing and Most Frequent Queries information gathered?](#) - Solution #00000412
- [How does SQL Diagnostic Manager gather statistics from monitored SQL Servers?](#) - Solution #00000055
- [How does SQL Diagnostic Manager calculate the reorganization percentage for database tables?](#) - Solution #00000301
- [What does the 'Memory Usage' metric in the console and reports represent?](#) - Solution #00002237


Deploy SQL Diagnostic Manager reports to Reporting Services

In addition to the included reports, SQL Diagnostic Manager also allows you to generate reports using Microsoft Reporting Services. Microsoft Reporting Services allows you to build powerful, [custom reports to deploy](#) for a comprehensive auditing solution.

Access the Reports Deployment wizard

SQL Diagnostic Manager provides two paths to access the Reports Deployment wizard where you can select reports to deploy to a Microsoft Reporting Services Server. The first path is by clicking **Deploy them now** at the bottom of the **Reports > Getting Started** view. The second path is by first selecting a specific report in the **Reports > Getting Started** view, and then clicking the **Deploy Report** option from the Navigation Pane.

In order to use this feature, you must have access to a server with Microsoft Reporting Services. For more information about Reporting Services, see the Microsoft document, [Reporting Services Reports](#).

 In the case of interlinked reports such as the Availability Group Statistics and Topology reports, users need to make sure to first deploy both reports from SQL Diagnostic Manager or Microsoft Reporting Services displays an error message.

Select reports to deploy

SQL Diagnostic Manager includes built-in reports that are specially designed to generate commonly-requested metrics reports using the SQL Server data collected in the repository.

Select reports

In the [Reports Deployment](#) wizard check the box for each report you want to deploy to a Microsoft Reporting Services Server, then click **Next**.

Available Reports

You can generate reports from the following report categories:

Server Monitoring Reports

Report Name	Description
Enterprise Summary	View the health of your SQL Servers.
Server Summary	View the health details of a single SQL Server.
Active Alerts	View the active alerts for all monitored SQL Servers.
Mirroring Summary	View the health of your mirrored databases.
Metric Thresholds	View all metric thresholds for a server.
Availability Group Topology	View the current topology of your availability groups configuration.
Deadlock	View the deadlock trends across SQL Server database.

Virtualization Monitoring Reports

Report Name	Description
Virtualization Summary	View a summary of your virtualized environment.
Virtualization Statistics	Analyze the performance trends of your virtual machines.

Activity Monitoring Reports

Report Name	Description
Change Log Summary	View a summary of all key actions performed in your SQLdm configuration.

Server Analysis Reports

Report Name	Description
Top Servers	Identify your worst performing SQL Servers.
Server Statistics	Analyze and compare performance trends across two SQL Servers.
Server Inventory	Find SQL Servers that share the same properties.
Query Overview	Identify your worst performing queries.
Top Queries	Find queries that are performing poorly or executing frequently.
Alert History	Analyze the alert history for your SQL Servers.
Baseline Statistics	Analyze and compare performance baselines across two SQL Server instances.
Query Wait Statistics	Analyze different wait type categories on your SQL Server instance.

Database Analysis Reports

Report Name	Description
Top Databases	Identify your worst performing databases.
Database Statistics	Analyze and compare performance trends across two databases.
Top Database Applications	Find database applications that consume system resources.
Mirroring History	Analyze the event history for a mirrored database.
Top Tables by Growth	Identify the fastest growing tables.

Report Name	Description
Top Tables by Fragmentation	Identify the most fragmented tables.
Tempdb Statistics	Analyze the performance trends of your tempdb database.
Transaction Log Statistics	Analyze the database transaction log.
Availability Group Statistics	View the historical health of your availability groups, availability replicas, and availability databases. Plus view additional information through the Availability Group Database Statistics sub-report.

Resource Analysis Reports

Report Name	Description
Session Statistics	Track key session and network performance metrics over time.
CPU Statistics	Track key CPU performance metrics.
Disk Details	Track key disk metrics for server.
Disk Statistics	Track key disk performance metrics.
Replication Statistics	Track key replication performance metrics.
Memory Statistics	Track key memory performance metrics.
Disk Space Usage	Track the most recent snapshot of available disk space.
Disk Space History	Track the history of activity for your disk.

Plan Reports


Report Name	Description
Disk Space Usage	Predicts how much disk space may be needed in the future based on current and historical growth rate trends
Database Growth	Predicts how large selected databases may grow based on current and historical growth rate trends.

Report Name	Description
Table Growth	Predicts how large selected tables may grow based on current and historical growth rate trends.

Specify a reporting services server and data source

In order to successfully deploy the reports, you must specify a Microsoft Reporting Services Server and a data source for the report.

Type the name of the report server or select it from the drop-down list, and then click **Next**.

 You must type the full URL address of the Microsoft Reporting Services Server. For example, `http://<ReportingServer>/reportserver`.

Configure report email notifications

Enter the email address where you would like SQL Diagnostic Manager to send the report notification, and then click **Next**. For more information on configuring report servers for email delivery, see the Microsoft document, [How to: Configure a Report Server for E-mail Delivery \(Reporting Services Configuration\)](#).

View the report deployment summary

Review that all the Reporting Services settings are correct and click **Finish** to apply your changes. If you want to make any changes to the settings displayed, click the **Back** button to go back and make your changes.

Custom reports

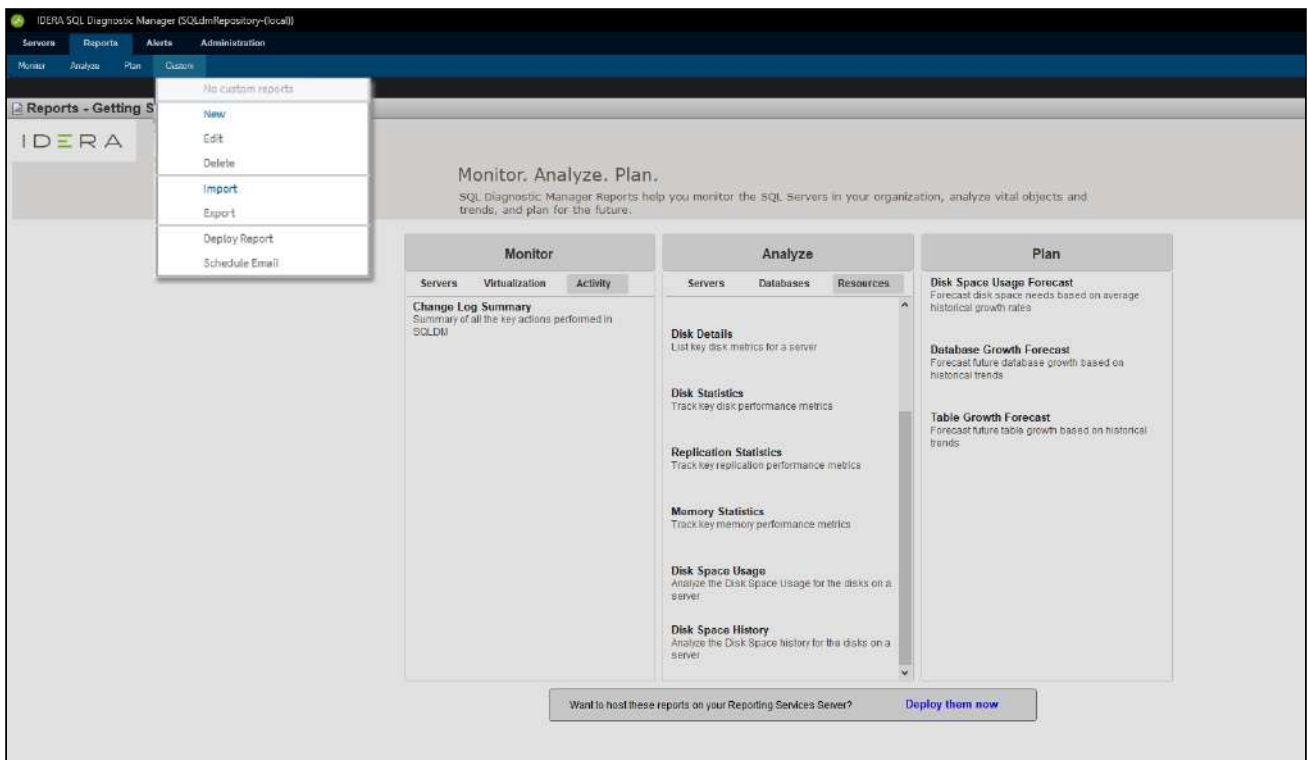
The Custom Reports view allows you to create, import or export reports on any counter collected by SQL Diagnostic Manager and on any of your [custom counters](#). You can [create custom reports](#) to view graphs of the counters followed by a grid containing all of the counters graphed over the entered time frame.

✔ Import and export custom reports

You can also [import](#) or [share](#) custom reports in the [IDERA community site](#).

Access Custom Reports

Once in the Reports view, click **Custom** in the Navigation pane.



Deploy reports to Microsoft Reporting Server

You can use the SQLDM Reports Deployment wizard to [deploy specified reports to your Microsoft Reporting Server](#).

Edit or Delete an existing report

You can edit or delete an existing custom report by selecting the report name from the list displayed in the Reports pane, and then click **Edit** or **Delete**. To edit your report, use the Custom Report wizard.

Import or Export an existing report

You can [import](#) or [export](#) an existing custom report to the [IDERA community site](#).

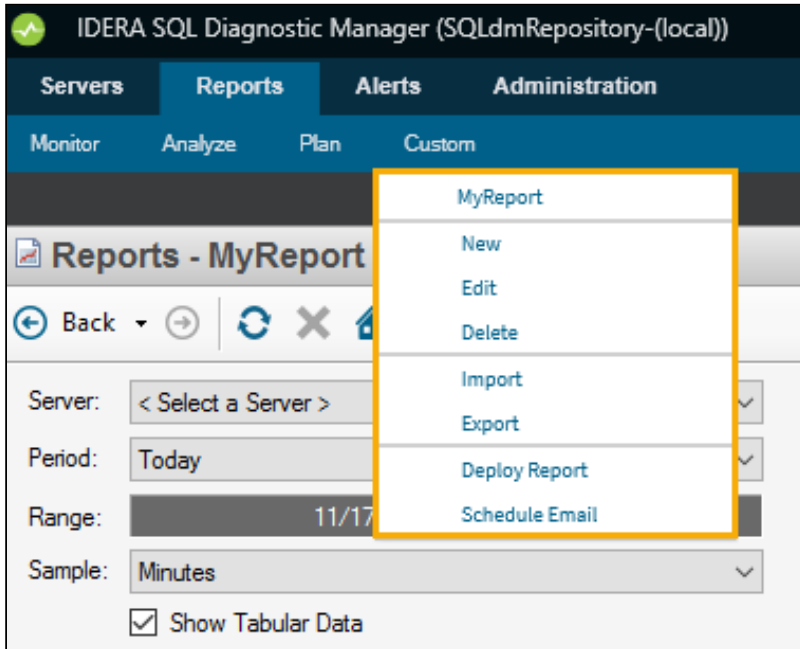
Create a custom report

The Custom Reports wizard allows you to create or edit a custom report. Custom Reports can include any metric collected by SQL Diagnostic Manager. This wizard allows you to choose the counters you want to include in your report, order the way the metrics appear, and specify the aggregation method used on each of your metrics.

To open the Custom Reports wizard, click the **New** option in the drop-down list under the Reports tab.

Add or edit a custom report

On the Custom drop-down list, you can select whether you want to create a new report or edit an existing report.





Create a new Report

To create a new report, select **New** from the drop-down list. When the Custom Report Wizard window opens, click **Next**, type a name for the report, and then click **Next**.

Add Custom Report ? X

Select the counters that you would like to include in your custom report.
Choose a counter type and the specific counter that you would like to report on



 SQL Diagnostic Manager gathers Windows and SQL Server counters as well as the custom counters that you have created. Please choose which of these counters you would like to include in this custom report.

Report Type

Show Graphical Data
 Include tabular data in report
 Show worst performing servers for each metric.

Available Counters

Operating System Counters
 SQL Server Counters
 Virtualization Counters
 Custom Counters

Pages/s
Processor Queue Length
Processor Time (%)
Total Byte VM
Used Byte VM
User Time (%)


Report Name: MyReport


Privileged Time (%)

To add the custom counters you want to include in your report, select them from the Available Counters list And and then click **Next**.

Add Custom Report ? X

Select the counters that you would like to include in your custom report.
Choose a counter type and the specific counter that you would like to report on



 SQL Diagnostic Manager gathers Windows and SQL Server counters as well as the custom counters that you have created. Please choose which of these counters you would like to include in this custom report.

Report Type

Show Graphical Data
 Include tabular data in report
 Show worst performing servers for each metric.

Available Counters

Operating System Counters
 SQL Server Counters
 Virtualization Counters
 Custom Counters

Pages/s
 Processor Queue Length
 Processor Time (%)
 Total Byte VM
 Used Byte VM
 User Time (%)


Report Name: MyReport

Privileged Time (%)

The Custom Report Wizard gives you the chance to order your custom counter according to your preference. Click **Next** when you are done.

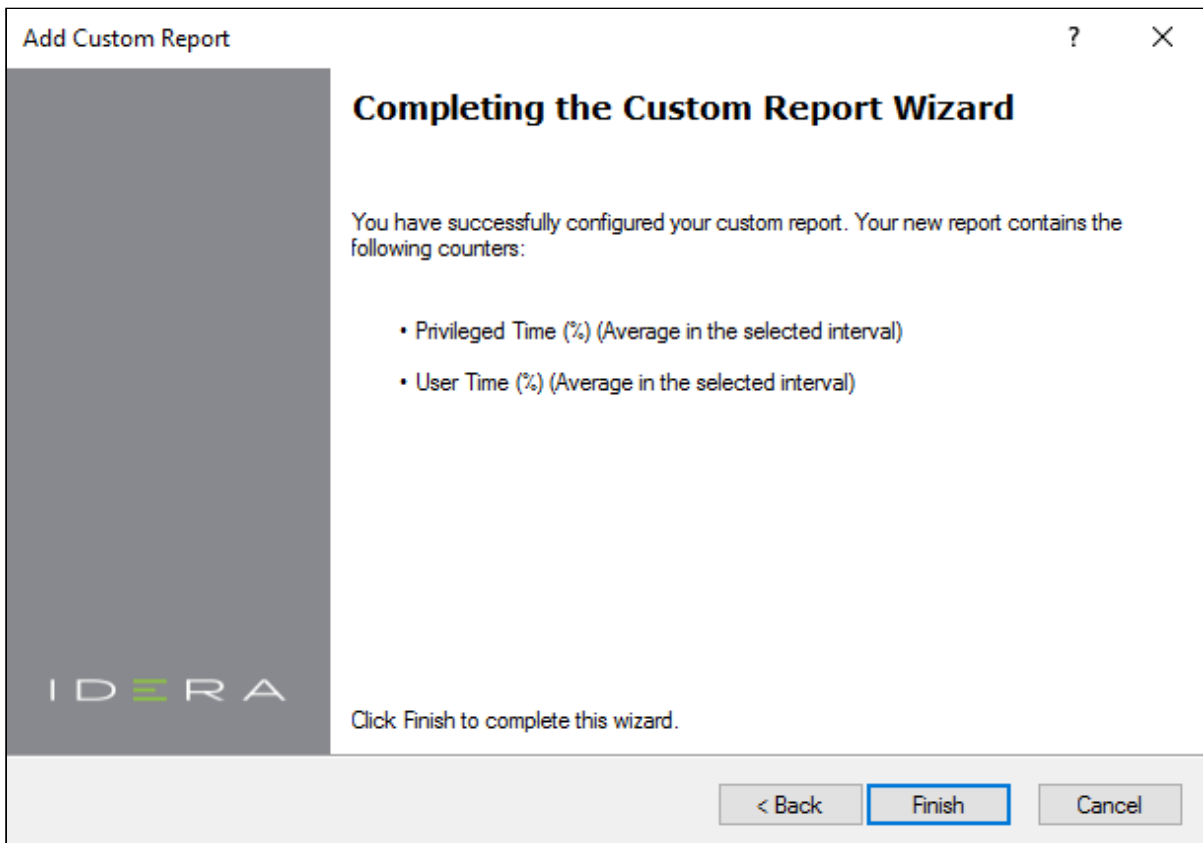
Add Custom Report ? X

Select counters for this report
Order and select aggregation for the counters that you have selected for this report.



Graph Title	Aggregation	Source	MoveUp	MoveDown
Privileged Time...	Weighted Avg	1	⊕	⊖
User Time (%)	Weighted Avg	1	⊕	⊖

To finish with your report check your custom counters summary and click **Finish**.



Edit a report

To edit an existing report, select the custom report from the Custom drop-down list. When the report opens, select **Edit** from the Custom drop-down list and edit your report.

Select report type

SQL Diagnostic Manager allows you to choose the type of custom report you want to create. You can select the show graphical data report or the [show worst performing servers for each metric](#) report.

If you want SQL Diagnostic Manager to display tabular data, select the **Include tabular data in report** check box.

- ✔ Enable the **Include tabular data in report** check box by clicking the **Show Graphical Data** radio button.

Select counter type and counters

The Select Counters window allows you to choose the counters you want to include in your report. You can select up to 10 counters from all of the counters that SQL Diagnostic Manager currently collects.

To select the counters you want to include in your custom report:

1. Check the box next to the type of counters you want to include with your report to view an alphabetical list of counters. You can select counters from the following groups:
 - a. Operating System Counters
 - b. SQL Server Counters
 - c. Virtualization Counters for both Hyper-V and VMware server virtualization platforms
 - d. Custom Counters
3. Select one or more counters from the Available Counters list and click **Add** to include it in your report.
4. Click **Next**.

Top Servers custom report

The Top Servers custom report lists your worst performing monitored SQL Server instances for each selected metric. The Top Servers custom report allows you to filter information by tag, time period, and number of servers.

When to run this report

Run the Top Servers custom report to quickly identify your monitored SQL Server instances that need immediate attention.

Select counter order

Update the counter order for this report to specify where you want SQL Diagnostic Manager to display the counters included in your custom report and how to aggregate metrics.

SQL Diagnostic Manager offers the following options to change the aggregation for each metric included in your report.

Maximum

Displays the maximum value for the metric since the last collection.

Average

Displays the average per second value since the last collection.

Per Minute


Displays the average per minute value since the last collection.

To rearrange or change the metric aggregation for your counters:

1. ***If you want to rearrange a counter in your report***, click the up or down arrow next to the counter you want to move.
2. ***If you want to change the aggregation for a metric in your report***, select the appropriate **Aggregation** method.
3. Click **Next**.
4. Complete the Custom Report wizard by clicking **Finish**.


Import a custom report

The Custom Reports Import Wizard allows you to import already defined custom reports from the [IDERA Community Site](#). Custom reports can include any metric collected by SQL Diagnostic Manager.

 The Custom Reports Import and Export functionality applies only to reports created within SQL Diagnostic Manager. Also you can export or import these reports only within SQL Diagnostic Manager environments.

To import a custom report:

1. Open the Custom Reports Import Wizard by selecting the **Import** button from the Custom reports drop-down list.


 You can also access to the Custom Report import wizard by clicking **Administration > Import/Export** from the navigation pane and selecting **Custom Report** in the import column.

2. On the **Welcome to the Custom Reports Import** page, click **Next**.
3. On the **Select custom reports to import** page, browse for custom reports' XML files to import.
4. If the report name is already in use but you wish to complete the import process, click **Replace existing report**.
5. If the imported custom report is valid. Click **Finish** to complete the import process and exit the Custom Reports Import Wizard.


 **Export a custom report**
For instructions on exporting a custom report, see [Export a custom report](#).

Export a custom report

The Custom Reports Export Wizard allows you to export and share already created custom reports in the [IDERA community site](#). Custom Reports can include any metric collected by SQL Diagnostic Manager.

 The Custom Reports Import and Export functionality applies only to reports created within SQL Diagnostic Manager. Also you can export or import these reports only within SQL Diagnostic Manager environments.

To export a custom report:

1. Select **Reports** in the navigation pane.
2. Once in the Reports view, click **Custom** in the Navigation pane.
3. Select the custom report you wish to export from the list and click **Export** in the Custom Report Options.  Click image to view full size.
4. In the **Save As** dialog box, indicate in which folder you want to save the custom report XML file.
5. In the "Selected custom report exported successfully" notification, click **OK**.

Import a custom report

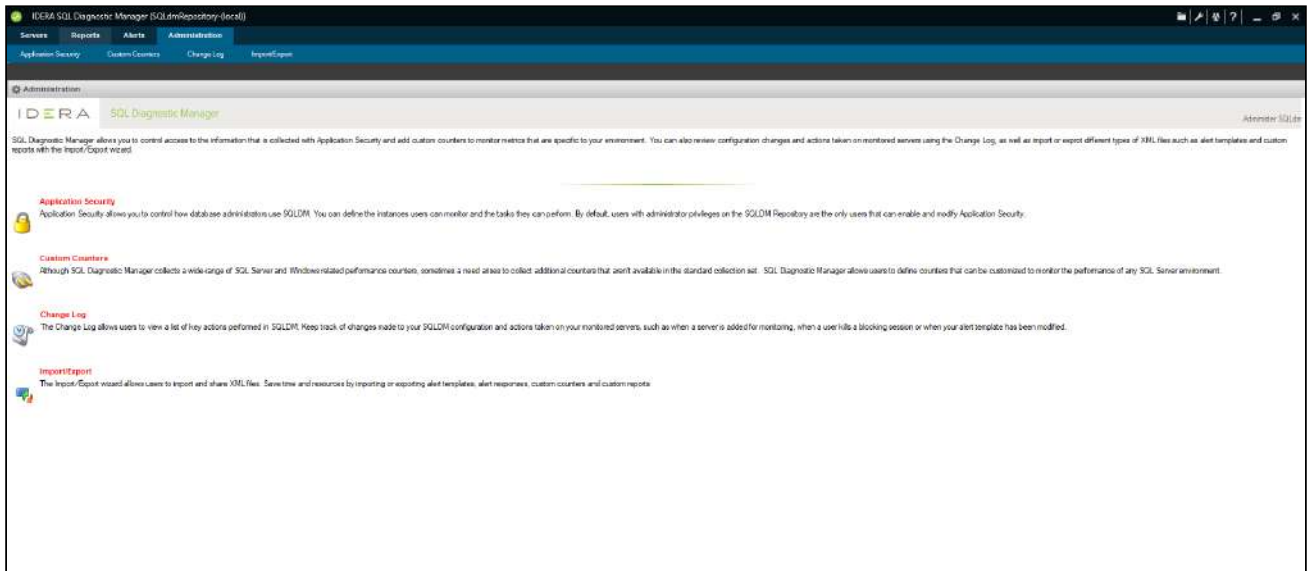
For instructions on importing a custom report, see [Import a custom report](#).

To export all your custom reports:

1. Select **Administration > Import/Export** and click **Custom Report** in the export column.
2. In the **Browse For Folder** dialog box, indicate in which folder you want to save the custom reports XML files.
3. Once you receive the notification "Custom reports exported successfully to selected directory", click **OK**.


Manage SQL Diagnostic Manager

The Administration view of SQL Diagnostic Manager allows you to [control access to all the information that SQL Diagnostic Manager collects](#), letting you designate the servers and databases that each of your SQL Diagnostic Manager users can view or edit. In addition to controlling access, you can also [add](#) or [import](#) custom counters to monitor in SQL Diagnostic Manager and even [keep track of key actions performed in your SQL Diagnostic Manager environment](#).



Use application security to manage SQL Diagnostic Manager

In many SQL Server production environments, there are several different types of users, such as DBAs, Operators, Help Desk Users, Administrators, and Auditors to name a few. Within each of these groups, there may be several sub-groups of responsibility, such as a DBA that manages a certain set of high security databases that other DBAs do not. SQL Diagnostic Manager allows you to designate the access both individual users and groups of users have within SQL Diagnostic Manager.

 When Application Security is enabled, the only users who have access to SQL Diagnostic Manager are those who belong to the sysadmin role on the SQL Diagnostic Manager Repository. All other users must be added to SQL Diagnostic Manager.

Enable Application Security

Once you enable Application Security, you must add User Permissions for each user you want to give SQL Diagnostic Manager access. Based on the assigned permissions, each user can only access a subset of monitored SQL Server instances. These security settings are enforced in the SQL Diagnostic Manager Console, the IDERA Newsfeed, and SQL Diagnostic Manager Mobile.

Note that users who belong to the sysadmin role on the SQL Diagnostic Manager Repository always have Administrator powers.

Available actions on the Application Security window

The Application Security window lists each login name and the type of access the login has to SQL Diagnostic Manager. The following actions are available on the Application Security window:

Disable/Enable Application Security

You can toggle Application Security on and off using the associated icon. Disabling Application Security gives all SQL Diagnostic Manager users full access to SQL Diagnostic Manager.

Add SQL Diagnostic Manager Permissions

You can open the Add SQL Diagnostic Manager Permissions wizard by clicking the **Add** button on the Application Security window. This wizard allows you to add SQL Server Logins, Windows Users and Group accounts and assign them SQL Diagnostic Manager view, modify, and administration privileges.

Edit Login Permissions

Once you add SQL Diagnostic Manager Permissions, the Edit Login Permissions window allows you to edit login permissions.

Delete Login Permissions

Click the **Delete** button to remove the previously added login from SQL Diagnostic Manager.

Enable/Disable Permissions


You can toggle non-system, SQL Diagnostic Manager login permissions by checking and clearing the associated box in the **Enabled** column.

Export SQL Diagnostic Manager Permission Settings to Excel


You can export all your SQL Diagnostic Manager Permission settings to Microsoft Excel by right-clicking any user in on the Application Security window and selecting **Export to Excel**.

Add users to SQL Diagnostic Manager

The Add Permission wizard allows you to give SQL Diagnostic Manager access to a specified login in your environment. By default, all users belonging to the sysadmin role on the SQL Diagnostic Manager Repository have SQL Diagnostic Manager Administrator privileges and all other users are denied access to any of the monitoring and diagnostic views in SQL Diagnostic Manager.

 Logins belonging to the sysadmin role on the SQL Diagnostic Manager Repository have SQL Diagnostic Manager Administrator privileges and you cannot edit or delete these accounts.

Each login account can have multiple permissions associated with it. For example, if you want User A to have the ability to view all your SQL Server instances but modify only one of them, you can create two permissions. Include View permissions in the first login to allow the login to view all your SQL Server instances, and enable Modify permissions for the specified SQL Server instance in the second login.


 If a SQL login is entered and it does not already exist, a new login account is created on the SQL Diagnostic Manager Repository SQL Server instance.

To create a SQLDM login account:

1. Click **Administration > Application Security**.
2. Click the **Add** button on the Application Security window.
3. Click **Next** on the Add Permission wizard Welcome window.
4. Type either the Windows account or SQL login used to connect to the SQL Diagnostic Manager Repository.
5. Select the **Authentication** type and click **Next**.
6. Select the [type of SQL Server instance permissions](#) to grant to the specified account and click **Next**.
7. Select the tags or SQL Server instances to apply the permissions to and click **Add** to move them to the Selected Servers column and click **Next** (this option is only available for View and Modify permissions only).
8. Review your changes on the Summary window and click **Finish** to add the SQL Diagnostic Manager login.

Specify user

The Specify User window allows you to enter the login name and authentication type. Enter either the Windows account name or the SQL login used to connect to the SQLDM Repository. If the SQL login entered does not exist, one is created on the SQLDM Repository Server.

 If the Windows account entered is not a valid Windows account, an error appears when you exit the Add Permission wizard and the SQLDM login is not created.

Specify permissions

In this window you can specify the SQLDM permissions you want to assign to the login. You can apply the following permissions to the specified account:

- View data collected for monitored SQL Server instances
- Modify configuration and view data collected for monitored SQL Server instances
- View data, acknowledge alarms, and control maintenance mode status
- Administrator powers in SQL Diagnostic Manager

- i** Give access to the Web Application to the specified account, by selecting the Grant Web Application Access option.

View Permissions

Logins with View permissions can view the data collected for the assigned SQL Server instances except where noted in the following table.

SQLDM Area	Cannot do the following
File Menu	Manage servers
Tools Menu	Configure default alerts
	Edit notification rules or providers
	Edit grooming options
Help Menu	Manage licenses
Servers pane	Delete servers
	Enter or exit Maintenance Mode
	Configure alerts, but can view alert settings
	Edit SQL Server properties, but can view properties
Today Page > Active Alerts	Configure alerts, but can view alerts
Today Page > Common Tasks	Manage servers
Mini-Monitors grid	Delete mini-monitors
	Edit properties, but can view properties
Server Overview > Dashboard	Enter or exit Maintenance Mode
	Configure alerts, but can view alert settings
Server Overview > Details	Configure alerts, but can view alert settings
Server Overview > Configuration	Edit value
Session > Details	Trace or kill a session
Session > Locks	Trace or kill a session

SQLDM Area	Cannot do the following
Session > Blocking	Trace or kill a session
Queries/Non-query activities	Configure Query/Activity Monitor properties, but can view properties
Resources > Procedure Cache	Clear cache
Databases > Tables and Indexes	Update statistics or rebuild indexes
Services > Summary	Start or stop services
Services > SQL Agent Jobs	Start or stop job
Logs	Cycle server log
	Configure
Alerts pane	Configure default alerts
	Edit notification rules and providers
Alerts > Grid	Configure alerts, but can view alerts
	Clear alerts
	Clear alerts of this type for this instance
Administration pane	Access or view Administration

Modify Permissions

Logins with Modify permissions can edit basic settings for the data collected for assigned SQL Server instances except where noted in the following table.

SQLDM Area	Cannot do the following
File Menu	Manage servers

SQLDM Area	Cannot do the following
Tools Menu	Edit alert configuration
	Edit notification rules or providers
	Edit grooming options
Help Menu	Manage licenses
Alerts pane	Configure default alerts
	Edit notification rules or providers
Administration pane	Access or view Administration

Administration Permissions

Logins with Administration permissions can access all monitored SQL Server instances and modify all settings including grooming options, notification rules, and providers, manage SQL Diagnostic Manager licenses, manage application security including the ability to add custom counters, and has the ability to add and remove SQL Server instances in SQL Diagnostic Manager.

- ✔ You cannot edit or delete logins belonging to the sysadmin role on the SQL Diagnostic Manager Repository as they have SQL Diagnostic Manager Administrator privileges.

- ℹ The SQLDM Console displays a message when a user with View or Modify permissions attempts to perform an action that requires Administrator (sysadmin) permissions.

Assign instances to user


The Select SQL Server window allows you to select the SQL Server instances to apply the permissions you set on the Specify User Permissions window. Select [Server tags from the Tags drop-down list](#) or [choose New tag from the list](#). Select the SQL Server Instances you want the login to have permissions on and click **Add** button to move them to the Selected Servers column. Click **Next** to continue.

View the Add Permission summary

Review that all the Add Permission settings are correct and click **Finish** to apply your changes. If you want to make any changes to the settings displayed, click the **Back** button to go back and make your changes.

Edit user permissions

The Edit Permissions window allows you to change the permissions for the specified login.

 Logins belonging to the sysadmin role on the SQLDM Repository have SQLDM Administrator privileges and you cannot edit or delete the login.

To Enable/Disable Login Permissions

You can toggle non-system, SQLDM login permissions by checking and clearing the associated check box in the **Enabled** column.

Edit Permissions

Select the [appropriate permission level](#) for the login. You can select **View**, **Modify**, or **Administrator** permissions.

Enter Comments

The **Comments** field allows you to enter additional information about the login. This information appears in the **Comment** column on the Application Security window.

Edit assigned instances

The Assigned Servers window allows you to change the SQL Server instances on which the login has permissions. To [tag a server instance](#), select [Server tags from the Tags drop-down list](#) or add a new tag by [choosing New tag from the list](#). Select the SQL Server instances on which you want the login to have permissions and click **Add** button to move them to the **Selected Servers** column. Click **Next** to continue.

Use the Change Log to review changes in your SQL Diagnostic Manager configuration

Granted the number and different types of users in SQL Server production environments, keeping track of changes to your SQL Diagnostic Manager configuration is critical. Through the Change Log you can view a list of key actions performed in your SQL Diagnostic Manager environment, such as when a server is added for monitoring, an alert configuration or grooming configuration is changed. The Change Log view gives thorough and instant information on modifications.

The Change Log view is available by default in SQL Diagnostic Manager version 8.0 and higher versions.

Access the Change Log view

To open the Change Log view, click **Administration > Change Log** from the navigation pane.

Change Log Window Properties

This window allows you to view details about an individual event in the Change Log. You can view the following information:

- Date and time of action
- Type of action
- Object on which the action occurred
- Workstation and workstation user
- Repository user who executed the action
- Description of action

Available actions

Scroll through activities

To scroll from one action to the next, use the up and down arrows.

Copy, export and print action details

To copy action details to another application, export contents to Microsoft Excel or even print, right-click on selected row. This action prompts the aforementioned commands.

Getting more information

You might be interested in obtaining further details about the actions performed in SQL Diagnostic Manager and their possible impacts. The details pane contains in depth action information and automatically displays when a row in the table is highlighted. The split bar is adjustable to your needs.

Filters

Allows you to filter the listed activities by time span, for example, last 30 days or even arrange a custom range.

Enable Groups

Allows you to group actions by a specific header column, such as the workstation from which an action was executed or the date an action occurred. To enable groups, click Group By Box. You can as well expand all groups or collapse all groups.

Available columns

Date

Provides date and time when action occurred.

Action

Provides information on the type of action that occurred.

Object Name

Provides the name of the object on which the action was executed.

Workstation

Provides the name of the workstation from which an action was executed.

Workstation User

Provides the name of the user, logged in a specific workstation, from which an action was executed.

Repository User


Provides the name of the user who applied the change. The format displayed, when using Windows authentication, is DomainName\LogonName.

Change Summary

Provides a brief description of action.

Use the SQL Diagnostic Manager import/export wizard

SQL Diagnostic Manager provides a centralized import/export wizard that allows you to import and share existing XML files. Save time and resources by importing or exporting alert templates, alert responses, custom reports, and custom counters files.

 The import/export wizard is available by default in SQL Diagnostic Manager 10.0 and higher versions.

Access the import/export wizard

To open the import/export wizard, click **Administration > Import/Export** at the navigation pane.

Available options in the import/export wizard

Alert Template

Click **Alert Template** in the appropriate column to [import](#) or [export](#) an alert template.

Alert Response

Select **Alert Response** in the appropriate column to [import](#) or [export](#) an alert response.

Custom Report

Click **Custom Report** in the appropriate column to [import](#) or [export](#) a custom report.

Custom Counters

Select **Custom Counter** in the appropriate column to [import](#) or [export](#) a custom counter.

Use PowerShell to automate SQL Diagnostic Manager functions

SQL Diagnostic Manager PowerShell integration allows you to make mass changes to your SQL Diagnostic Manager configuration across multiple SQL Server instances in a few lines of text, saving you time. For more information about it click the following link: [SQL Diagnostic Manager PowerShell integration](#).

About PowerShell

Microsoft® Windows® PowerShell is a task-based command-line shell and scripting language designed especially for system administration. Built on the .NET Framework, Microsoft® Windows® PowerShell helps IT professionals and power users control and automate the administration of the Windows operating system and applications that run on Windows.


Windows PowerShell furnishes an operating environment for commands that include cmdlets, functions, filters, scripts, aliases, and executables (applications). The main command type used in this environment is the cmdlet, with certain cmdlets made available only through Windows PowerShell providers that allow access to stored data.

Windows PowerShell operates within a hosting application (the default is powershell.exe) that exposes a command line to the user, and uses a host interface to communicate with the commands invoked by the command line. You can use a console application, Windows application, or a Web application as the hosting application. In most cases, the hosting application uses its Main function to interact with the Windows PowerShell runtime through the internal host interface; however, a hosting application can optionally support its own custom host by implementing the PSHost class along with one or more related user interface classes. Together, these classes allow direct communication between the application and Windows PowerShell commands.

For more information about Windows PowerShell, see [SQL Server Books Online](#).

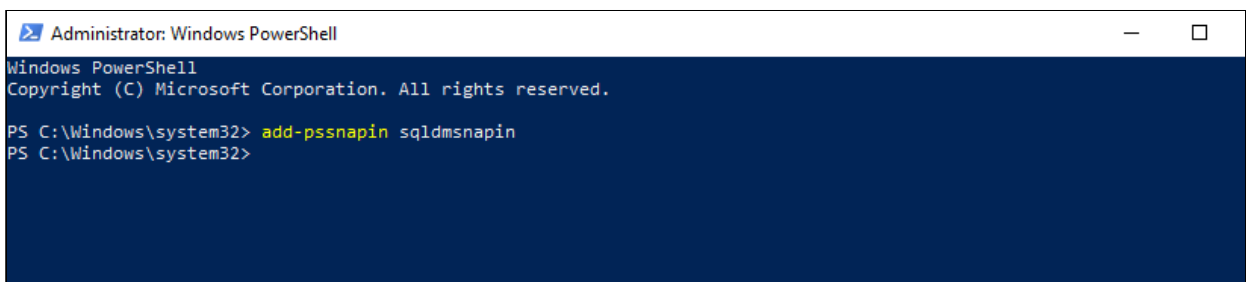
Install the SQL Diagnostic Manager provider for PowerShell

The SQLDM Provider is installed automatically with SQL Diagnostic Manager. To use SQL Diagnostic Manager snapin, you have to enable the provider and register the SQLDM drive.

 PowerShell 3.0 or higher is required to use the PowerShell snap-in.

To enable the provider, type the following within PowerShell:

```
add-pssnapin sqldmsnapin
```



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> add-pssnapin sqldmsnapin
PS C:\Windows\system32>
```

To verify your snapin is registered:

```
get-psprovider
```

```

PS C:\Windows\system32> get-psprovider

Name                Capabilities                Drives
----                -
Registry            ShouldProcess, Transactions {HKLM, HKCU}
Alias               ShouldProcess              {Alias}
Environment         ShouldProcess              {Env}
FileSystem          Filter, ShouldProcess, Credentials {C, D}
Function            ShouldProcess              {Function}
Variable            ShouldProcess              {Variable}
SQLdm               Credentials                {}

```

To register the SQLdm drive, type the following within PowerShell:

```
New-SQLdmDrive dm [Server] [Repository]
```

For example:

```
New-SQLdmDrive dm PUL-FF-W2K8X64S\SQL2012_BI SQLdmRepository
```

Creates a new SQL Diagnostic Manager drive called 'dm' that is connected to the SQLdmRepository database on the PUL-FF-W2K8X64S\SQL2012_BI SQL Server using Windows Authentication.

```

PS C:\Windows\system32> New-SQLdmDrive dm DESKTOP-IF6DH7B SQLdmRepository

Name        Used (GB)  Free (GB)  Provider  Root        CurrentLocation
----        -
dm          -----  -----  SQLdm     dm:\

```

✓ PowerShell help

To receive general help, type the following within PowerShell:

```
get-help
```

To display a list of cmdlets for the SQL Diagnostic Manager snapin:

```
get-help sqldm
```

Applications available to use with PowerShell

There are several applications that allow you to use PowerShell and enhance its functionality.

PowerShell Plus is a powerful interactive scripting environment designed to help administrators and developers learn and master Windows PowerShell. Unlike most of the basic script editors on the market today, PowerShell Plus features a powerful interactive console, an advanced script editor and a comprehensive learning center. PowerShell Plus is built on the backbone of PowerShellIDE, the first PowerShell application to offer a very rich graphical user interface.

Interactive PowerShell Console

The PowerShellPlus Console allows you to work interactively with PowerShell from a feature rich Windows UI. This integration makes working with PowerShell faster and easier to use than ever before.

Advanced Script Editor


The advanced debugger and script editor lets you build and test complex PowerShell scripts, try one line PowerShell commands from an embedded console window, and sign your script with a security certificate, all from a single workspace.

Comprehensive Learning Center

The Comprehensive Learning Center helps you experience PowerShell by example. Short tutorials guide you through basic concepts at your own pace. The Comprehensive Learning Center also includes dynamically created help topics from currently installed PowerShell CmdLets.

PowerShell Cmdlets for SQL Diagnostic Manager

The following cmdlets help you automate the administration of your SQL Diagnostic Manager deployment.

 To use SQL Diagnostic Manager snapin, you have to [enable the provider and register the SQLDM drive](#).


Cmdlet Name	Description	Examples
Escape-SQLdmName	Allows you to specify server names containing special characters such as <code> \?/<>[]:</code> *	<pre>Escape-SQLdmName Server\Instance</pre> <ul style="list-style-type: none"> Escapes the instance name <code>Server\Instance</code>. <pre>Escape-SQLdmName Server\Instance1,Server\Instance 2</pre> <ul style="list-style-type: none"> Escapes the array of instance names. <pre>Escape-SQLdmName Server\Instance Escape-SQLdmName -undo</pre> <ul style="list-style-type: none"> Escapes the instance name <code>Server\Instance</code> and then undo the operation. <pre>Set-SQLdmMonitoredInstance (Escape-SQLName Server\Instance1) - WindowsAuthentication</pre> <ul style="list-style-type: none"> Escapes the instance name for a Windows user.
Get-SQLServers	Retrieves a list of SQL Servers on the network	<pre>Get-SQLServers</pre> <ul style="list-style-type: none"> Returns a list of all known SQL Servers. <pre>Get-SQLServers -localonly</pre> <ul style="list-style-type: none"> Returns a list of SQL Servers running on the local machine.

Cmdlet Name	Description	Examples
Grant- SQLdmPermissions	Grants SQL Diagnostic Manager permissions on a server	<p>Grant-SQLdmPermission -Path dm:\appsecurity\AUser\AServer -Permission View</p> <ul style="list-style-type: none"> Grants View permissions to instance 'AServer' to user 'AUser'. <p>Grant-SQLdmPermission -Path("dm:\appsecurity\" + (Escape-SQLdmName -Name MyDomain\AUser)) -Name AServer -Permission Modify</p> <ul style="list-style-type: none"> Grants Modify permissions to instance 'AServer' to Windows user 'MyDomain\AUser'. <p>Grant-SQLdmPermission -Path dm:\appsecurity\AUser -Tag Prod -Permission Modify</p> <ul style="list-style-type: none"> Grants a user permissions to instances tagged as Prod.
New- SQLdmAlertTemplate	Associates a template to a monitored instance or a tag	<p>New-SQLdmAlertTemplate</p> <ul style="list-style-type: none"> Enter the following parameters when using this command: TemplateName, DataSource, and RepositoryName or Tag. For example: <div data-bbox="995 1285 1423 1720" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>New-SQLdmAlertTemplate</p> <pre>PS dm:\> New-SQLdmAlertTemplate -DataSource DESKTOP-IF6DHB -RepositoryName SQLdmRepository -RepositoryInstances DESKTOP -IF6DHB -TemplateName "Default Template"</pre> </div>

Cmdlet Name	Description	Examples
New-SQLdmDrive	Creates a drive for connecting to the SQLdm Repository	<p>New-SQLdmDrive dm MYSQLDMSERVER SQLdmRepository</p> <ul style="list-style-type: none"> Creates a new SQL Diagnostic Manager drive called 'dm' that is connected to the SQLdmRepository database on the MYSQLDMSERVER SQL Server using Windows Authentication. <pre>New-SQLdmDrive dm MYSQLDMSERVER SQLdmRepository -Credential sa</pre> <ul style="list-style-type: none"> Creates a new SQL Diagnostic Manager drive called 'dm' that is connected to the SQLdmRepository database on the MYSQLDMSERVER SQL Server using SQL Authentication. Note that Powershell will prompt for the password.
New-SQLdmMonitoredInstance	Adds a SQL Server Instance to monitor in SQL Diagnostic Manager	<pre>New-SQLdmMonitoredInstance -Path Instance1 -WindowsAuthentication -tags production,finance,critical</pre> <ul style="list-style-type: none"> Adds a new instance named Instance1 to SQL Diagnostic Manager. <pre>New-SQLdmMonitoredInstance -Path (Escape-SQLdmName -Name ServerA\Instance1) -Credential sa</pre> <ul style="list-style-type: none"> Adds a new instance named ServerA\Instance1 to SQL Diagnostic Manager.

Cmdlet Name	Description	Examples
New-SQLdmUser	Creates a new SQL Diagnostic Manager user	<pre>New-SQLdmUser -Path fred -SQLLogin -Password xxxx -Comment "Fred is an admin"</pre> <ul style="list-style-type: none"> Creates a new user called fred that uses SQL Authentication. Note that if the password is not specified, PowerShell will prompt for the password. <pre>New-SQLdmUser (Escape-SQLdmName -Name mydomain\fred) -WindowsUser</pre> <ul style="list-style-type: none"> Creates a new user called mydomain\fred that uses Windows Authentication.
Remove-SQLdmMonitoredInstance	Removes a SQL Server Instance from SQL Diagnostic Manager monitoring	<pre>Remove-SQLdmMonitoredInstance -Path \Instances\ServerA</pre> <ul style="list-style-type: none"> Removes the instance named ServerA from SQL Diagnostic Manager. <pre>Remove-SQLdmMonitoredInstance (Escape-SQLdmName -Name ServerA\Instance1) -retaindata</pre> <ul style="list-style-type: none"> Removes the instance named ServerA from SQL Diagnostic Manager but retains its data for reporting.
Remove-SQLdmUser	Removes an existing SQL Diagnostic Manager user	<pre>Remove-SQLdmUser -Path / AppSecurity/fred</pre> <ul style="list-style-type: none"> Removes user fred from SQL Diagnostic Manager. <pre>Remove-SQLdmUser (Escape-SQLdmName -Name MyDomain\AUser)</pre> <ul style="list-style-type: none"> Removes Windows user MyDomain\AUser from SQL Diagnostic Manager.

Cmdlet Name	Description	Examples
Revoke- SQLdmPermission	Revokes SQL Diagnostic Manager permissions on a server	<p style="text-align: center;">Revoke- SQLdmPermission -Path \AppSecurity\fred\ServerA</p> <ul style="list-style-type: none"> • Revokes user fred's permissions to instance ServerA. <p style="text-align: center;">Revoke-Permission -Path /AppSecurity/fred -Name ServerA</p> <ul style="list-style-type: none"> • Revokes user fred's permissions to instance ServerA. <p style="text-align: center;">Revoke-Permission -Path / AppSecurity/fred -Tag Prod</p> <ul style="list-style-type: none"> • Revokes user fred's permissions to instances tagged as Prod. <p style="text-align: center;">Revoke-SQLdmPermission -Path ("dm:\AppSecurity\" + (Escape- SQLdmName -Name MyDomain\UserA)) -Name (Escape-SQLdmName -Name ServerA\Instance1)</p> <ul style="list-style-type: none"> • Revokes Windows user MyDomain\UserA permission to ServerA\Instance1.

Cmdlet Name	Description	Examples
Set-SQLdmAppSecurity	Enables or disables Application Security for SQL Diagnostic Manager	<pre>Set-SQLdmAppSecurity \AppSecurity -Enabled</pre> <ul style="list-style-type: none"> Enables SQL Diagnostic Manager Application Security. <pre>Set-SQLdmAppSecurity \AppSecurity -Disabled</pre> <ul style="list-style-type: none"> Disables SQL Diagnostic Manager Application Security.
Set-SQLdmLogging	Sets the logging level of the SQL Diagnostic Manager provider	<div style="border: 1px solid red; padding: 5px; background-color: #ffe6e6;">  Use Set-SQLdmLogging for debugging purposes only. This cmdlet should be used at the request of IDERA Technical Support. </div>
Set-SQLdmMonitoredInstance	Configures options on your SQL Server Instance	<pre>Set-SQLdmMonitoredInstance -Path Instance1,Instance2,Instance3 -Credential sa</pre> <ul style="list-style-type: none"> Updates the credentials used for Instance1, Instance2, and Instance3 to use SQL Authentication and sets a new login and password. Note that PowerShell will prompt for the password. <pre>Set-SQLdmMonitoredInstance (Escape-SQLdmName -Name ServerA\Instance1) -WindowsAuthentication</pre> <ul style="list-style-type: none"> Updates the credentials used for ServerA\Instance1 to use Windows Authentication.


Cmdlet Name	Description	Examples
Set-SQLdmUser	Set the properties of an existing SQL Diagnostic Manager user	<pre>Set-SQLdmUser -path \AppSecurity\bozo -Disabled -Comment "Bozo is disabled."</pre> <ul style="list-style-type: none"> Disables the Set SQL Diagnostic Manager user bozo. <pre>Set-SQLdmUser fred -Enabled -Administrator</pre> <ul style="list-style-type: none"> Enables SQL Diagnostic Manager user fred and gives him administrator rights. <pre>Set-SQLdmUser (Escape- SQLdmName -Name mydomain\fred) -Disabled</pre> <ul style="list-style-type: none"> Disables the SQL Diagnostic Manager user mydomain\fred.

Customize your Monitored SQL Server properties within PowerShell

SQL Diagnostic Manager allows you to customize your monitored SQL Server properties within PowerShell.

With PowerShell you can:

- [Edit general settings](#)
- [Edit query monitor settings](#)
- [Edit query waits settings](#)
- [Edit activity monitor settings](#)
- [Exclude applications, databases, and SQL statements within PowerShell](#)
- [Disable replication collection settings](#)
- [Customize table statistics collection](#)
- [Customize maintenance mode settings](#)

 To use SQL Diagnostic Manager snap-in, you have to [enable the provider and register the SQLDM drive](#).

Edit general settings

Edit the general settings of your monitored SQL Server with the following PowerShell commands:

Cmdlet Name	Description	Examples
-FriendlyServerName	Sets friendly name to an instance.	SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name ServeA\Instance1) -FriendlyServerName stringName
-FriendlyServerName Blank	Clears the friendly name previously assigned to an instance.	SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name ServeA\Instance1) -FriendlyServerNameBlank
-InputBufferLimiter	Defines an Input Buffer Limiter value.	SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name ServeA\Instance1) -InputBufferLimiter 100
-InputBufferLimiter Enable	Enables the Input Buffer Limiter.	SetSQLdmMonitoredInstance Path (EscapeSQLdmName -Name ServeA\Instance1) -InputBufferLimiterEnable
-InputBufferLimiter Disable	Disables the Input Buffer Limiter.	SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name ServeA\Instance1) -InputBufferLimiterDisable

Cmdlet Name	Description	Examples
-addtag	Associates tags to a server. Note that any tags that are specified have to already exist.	Set-SQLDmMonitoredInstance -Path (Escape-SQLdmName -Name ServeA\Instance1) -addtag NewTag,Tag2
-removetag	Disassociates tags from a server. Note that any tags that are specified have to already exist.	Set-SQLDmMonitoredInstance -Path (Escape-SQLdmName -Name ServeA\Instance1) -removetag NewTag,Tag2
-ScheduledCollectionIntervalMinutes	Specifies how often (in minutes) scheduled data collection takes place.	Set-SQLDmMonitoredInstance -Path (Escape-SQLdmName -Name ServeA\Instance1) -ScheduledCollectionIntervalMinutes 10
-DatabaseStatisticsIntervalMinutes	Specifies how often (in minutes) database statistics collection takes place.	Set-SQLDmMonitoredInstance -Path (Escape-SQLdmName -Name ServeA\Instance1) -DatabaseStatisticsIntervalMinutes 120
-ExtendedSessionDataCollection	Enables or disables extended data collection.	Set-SQLDmMonitoredInstance -Path (Escape-SQLdmName -Name ServeA\Instance1) -ExtendedSessionDataCollection 1 <ul style="list-style-type: none"> • Enables extended data collection. Set-SQLDmMonitoredInstance -Path (Escape-SQLdmName -Name ServeA\Instance1) -ExtendedSessionDataCollection 0 <ul style="list-style-type: none"> • Disables extended data collection.

Cmdlet Name	Description	Examples
-Credential	Sets SQL Authentication credentials.	<pre>Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -Credential sa</pre> <ul style="list-style-type: none"> • Uses SQL authentication. Note that PowerShell will prompt for the password. <pre>Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -WindowsAuthentication</pre> <ul style="list-style-type: none"> • Uses the credentials of the SQLDM Collection Service to connect to the monitored instance when collecting statistics.
-EncryptConnection	Enables or disables encrypt connection.	<pre>Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -EncryptConnection 1</pre> <ul style="list-style-type: none"> • Enables encrypt connection. <pre>Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -EncryptConnection 0</pre> <ul style="list-style-type: none"> • Disables encrypt connection.

Cmdlet Name	Description	Examples
<code>-TrustServerCertificates</code>	Enables or disables trust server certificates.	<pre>Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -TrustServerCertificates 1</pre> <ul style="list-style-type: none"> Enables trust server certificates. Note that when specified with the <code>EncryptConnection</code> parameter, SQL Server will not validate the certificates sent from the client but the data will still be encrypted. <pre>Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -TrustServerCertificates 0</pre> <ul style="list-style-type: none"> Disables trust server certificates.

Edit query monitor settings

Customize the query monitor settings of your monitor SQL Server with the following PowerShell commands:

Cmdlet Name	Description	Examples
<code>-QMEnableTraceMonitoring</code>	Enables "Collect Query Data using SQL Trace".	<pre>SetSQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMEnableTraceMonitoring</pre>
<code>-QMEnableExtendedEvents</code>	Enables "Collect Query Data using Extended Events".	<pre>SetSQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMEnableExtendedEvents</pre>
<code>-QMEnableCollectQueryPlan</code>	Enables "Collect actual Query Plans".	<pre>SetSQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMEnableCollectQueryPlan</pre>
<code>-QMDisableCollectQueryPlan</code>	Disables "Collect actual Query Plans".	<pre>SetSQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMDisableCollectQueryPlan</pre>

Cmdlet Name	Description	Examples
-QMEnableCollectEstimatedQueryPlan	Enables "Collect Estimated Query Plans".	Set-SQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMEnableCollectEstimatedQueryPlan
-QMDisableCollectEstimatedQueryPlan	Disables "Collect Estimated Query Plans".	Set-SQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMDisableCollectEstimatedQueryPlan
-QMEnabled	Enables query monitoring.	Set-SQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMEnabled
-QMDisabled	Disables query monitoring.	Set-SQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMDisabled
-QMCaptureBatches	Captures SQL batches.	Set-SQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMEnabled -QMCaptureBatches 1
-QMCaptureStatements	Captures SQL statements.	Set-SQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMEnabled -QMCaptureStatements 1
-QMCaptureProcs	Captures stored procedure and triggers.	Set-SQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMEnabled -QMCaptureProcs 1
-QMQueryDuration	Sets up threshold Duration (in milliseconds).	Set-SQLDmMonitoredInstance -Path (EscapeSQLDmName -Name ServeA\Instance1) -QMQueryDuration 20000

Cmdlet Name	Description	Examples
-QMLogicalDiskReads	Sets up threshold Logical disk reads.	Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -QMLogicalDiskReads 20
-QMCpuUsage	Sets up threshold CPU Usage (in milliseconds).	Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -QMCpuUsage 500
-QMPhysicalDiskWrites	Sets up threshold Physical disk writes.	Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -QMPhysicalDiskWrites 20

Edit query waits settings

Customize the query waits settings of your monitor SQL Server with the following PowerShell commands:

Cmdlet Name	Description	Examples
-QWExtendedEnable	Enables "Use Extended Events (SQL Server 2012 +)".	SetSQLDmMonitoredInstance -Path (EscapeSQLDmName -Name INSTANCE_NAME) -QWExtendedEnable
-QWExtendedDisable	Disables "Use Extended Events (SQL Server 2012 +)".	SetSQLDmMonitoredInstance -Path (EscapeSQLDmName -Name INSTANCE_NAME) -QWExtendedDisable
-QWStatisticsDisable	Disables "Collect query wait statistics (SQL 2005+ only)".	SetSQLDmMonitoredInstance -Path (EscapeSQLDmName -Name INSTANCE_NAME) -QWStatisticsDisable
-QWStatisticsDuration	Enables "Collect query wait statistics at a specified time and duration".	SetSQLDmMonitoredInstance -Path (EscapeSQLDmName -Name INSTANCE_NAME) -QWStatisticsStartDate"yyyy- mm-dd-hh:mm" -QWStatisticsDuration mm:ss

Cmdlet Name	Description	Examples
<code>-QWStatisticsCollectIndefinite</code>	Enables "Collect query wait statistics with Collect indefinitely option".	<code>SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name INSTANCE_NAME) -QWStatisticsCollectIndefinite</code>

Edit activity monitor settings

Customize the activity monitor settings of your monitor SQL Server with the following PowerShell commands:

Cmdlet Name	Description	Examples
<code>-NQAEnableSQLTrace</code>	Enables "Use SQL trace".	<code>SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name SRISHTIPUROHIT\SQLEXPRESS) -NQAEnableSQLTrace</code>
<code>-NQAEnableExtendedEvent</code>	Enables "Use extended event (SQL 2012+)".	<code>SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name SRISHTIPUROHIT\SQLEXPRESS) -NQAEnableExtendedEvent</code>
<code>-NQADisableAutoGrow</code>	Disables "Capture Autogrow".	<code>SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name SRISHTIPUROHIT\SQLEXPRESS) -NQADisableAutoGrow</code>
<code>-NQAEnableAutoGrow</code>	Enables "Capture Autogrow".	<code>SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name SRISHTIPUROHIT\SQLEXPRESS) -NQAEnableAutoGrow</code>
<code>-NQAEnableCaptureBlocking</code>	Enables "Capture Blocking".	<code>SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name SRISHTIPUROHIT\SQLEXPRESS) -NQAEnableCaptureBlocking</code>
<code>-NQADisableCaptureBlocking</code>	Disable "Capture Blocking".	<code>SetSQLdmMonitoredInstance -Path (EscapeSQLdmName -Name SRISHTIPUROHIT\SQLEXPRESS) -NQADisableCaptureBlocking</code>

Cmdlet Name	Description	Examples
-NQACaptureBlocking	Sets "blocking process threshold".	SetSQLDmMonitoredInstance -Path (EscapeSQLDmName -Name SRISHTIPUROHIT\SQLEXPRESS) -NQACaptureBlocking 100

Exclude applications, databases and SQL statements within PowerShell

Exclude applications, databases, and SQL statements with the following PowerShell commands:

Cmdlet Name	Description	Examples
-QMExcludedApps	Excludes applications from the query monitor.	Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -QMExcludedApps excludedApp1, excludedApp2, excludedApp3
-QMExcludedDatabases	Excludes databases from the query monitor.	Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -QMExcludedDatabases DB1, DB2, DB3
-QMExcludedSql	Excludes SQL statements from the query monitor.	Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -QMExcludedSql "Select 1", "Select * from Table where ID=5"

Disable replication collection

Disable replication statistics collection with the following PowerShell command:

Cmdlet Name	Description	Example
-ReplicationStatisticsDataCollection	Disables replication statistics collection.	Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -ReplicationStatisticsDataCollection 0 <ul style="list-style-type: none"> • Disables replication statistics collection for server named ServerA\Instance1.

Customize table statistics collection

Customize table statistics collection with the following PowerShell commands:

Cmdlet Name	Description	Example
-QTStartTime	Specifies the time of day for quiet time collection to occur.	Set-SQLdmMonitoredInstance -Path (Escape-SQLdmName -Name ServeA\Instance1) -QTStartTime 02:30:00 -QTDays Monday,Tuesday -QTReorgMinTableSizeK 200 -QTExcludedDatabases Test
-QTDays	Specifies the days of the week to perform quiet time collection.	
-QTReorgMinTableSizeK	Specifies the quiet time minimum table size (in kilobytes) to collect reorganization statistics.	
-QTExcludedDatabases	Specifies the list of databases to exclude from quiet time data collection.	

Customize maintenance mode settings

Customize maintenance mode settings with the following PowerShell commands:

Cmdlet Name	Description	Examples
-MMNever	Sets maintenance mode to <i>Never</i> . Note that SQL Diagnostic Manager will collect data according to its normal scheduled collection interval.	Set-SQLdmMonitoredInstance -Path (Escape-SQLdmName -Name ServeA\Instance1) -MMNever
-MMAAlways	Sets maintenance mode to <i>Until further notice</i> . Schedule collection will not occur.	Set-SQLdmMonitoredInstance -Path (Escape-SQLdmName -Name ServeA\Instance1) -MMAAlways
-MMRecurring	Sets maintenance mode to <i>Recurring every week at the specified time</i> .	Set-SQLdmMonitoredInstance -Path (Escape-SQLdmName -Name ServeA\Instance1) -MMRecurring -MMModeDays Monday,Wednesday,Friday -MMModeStartTime 01:35:00 -MMModeDuration 01:00
-MMModeDays	Specifies the days of the week recurring maintenance mode will be effective.	
-MMModeStartTime	Specifies the start time for recurring maintenance mode.	

Cmdlet Name	Description	Examples
-MModeDuration	Specifies the length of time for recurring maintenance mode.	
-MModeStartDate	Specifies the start date and time for one-time maintenance mode.	Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name ServeA\Instance1) -MModeOnce -MModeStartDate "2013-03-25 15:00" -MModeEndDate "2013-03-25 15:30"
-MModeEndDate	Specifies the end date and time for one-time maintenance mode.	
-MModeOnce	Sets maintenance mode to <i>Occurring once at the specified time</i> .	

Example: Editing several properties at once within PowerShell

PowerShell allows you to set up several properties in one command-line. On the PowerShell window, use Set-SQLDmMonitoredInstance followed by the corresponding cmdlets of the properties you want to edit. For example, consider the properties listed below:

- Enable Query Monitoring
- Capture Batches
- No capture procedures
- Set Query duration at 200 (ms)
- Set Logical disk reads at 20
- Set CPU usage at 100
- Set Physical disk writes at 10
- Exclude Application: MyExcludedApp
- Exclude Databases: DB1, DB2, and DB3

You can customize these settings with the following PowerShell command-line:

```
Set-SQLDmMonitoredInstance -Path (Escape-SQLDmName -Name QA- Name
Server\Instance) -QMEnabled -QMCaptureBatches 1 -QMCaptureProc 0
-QMQueryDuration 200
-QMLogicalDiskReads 20 -QMCpuUsage 100 -QMPhysicalDiskWrites 10
-QMExcludedApps MyExcludedApp -QMExcludedDatabases DB1,DB2,DB3
```

Manage your SQL Diagnostic Manager license


SQL Diagnostic Manager requires a license key. This key allows you to access all SQL Diagnostic Manager features, including the IDERA Newsfeed and SQL Diagnostic Manager Mobile.

Though you can immediately begin using SQL Diagnostic Manager with the included evaluation key, consider updating the evaluation license as soon as possible. A registered license key allows you to unlock the evaluation time and registered instance limits so you can begin using SQL Diagnostic Manager to the fullest extent.

SQL Diagnostic Manager provides an intuitive interface for license key management. You can view the status of your license keys and add licenses to monitor additional instances. Each license allows you to monitor a predetermined number of SQL Server instances.

When you reach your license limit, SQL Diagnostic Manager does not let you add new instances. When your trial period expires, SQL Diagnostic Manager ceases to run.

To open the License Keys window, select  > **Manage Licenses** from the Toolbar menu.

 If you change the SQL Diagnostic Manager Repository to a different instance, make sure to request a new license key from IDERA Sales as this scenario is not supported.


Use the trial license


SQL Diagnostic Manager includes a limited-time trial account with every installation. Note that IDERA provides separate trial and production installers.

As the end of the trial period nears, SQL Diagnostic Manager warns you each time you start the console that your trial period is ending. At the end of this trial period, if the license is not upgraded, SQL Diagnostic Manager stops functioning.

SQL Diagnostic Manager limits the number of monitored SQL Server instances to 15 when using a trial license.


Upgrade your trial license to a full license

1. Contact [IDERA Sales](#) to upgrade your SQL Diagnostic Manager trial license.
2. Click  and select **Manage Licenses**.
3. Enter the new license key in the **New Key** section and click **Enter**. When prompted to type or copy to replace the current key, click **Yes** to continue.

 If after upgrading to a full license you desire to move your SQL Diagnostic Manager Repository to another instance, make sure to request a new license key from [IDERA Sales](#) as this scenario is not supported.

Add a new license

If you would like to monitor additional SQL Server instances beyond the limits of your existing license, contact [IDERA Sales](#) to purchase additional licenses.

Once you receive a new license, open the License Keys window, click , select **Manage Licenses**, type your license key in the available space, and click **Enter**. Your new license lists the number of additional SQL Server instances that you can monitor by SQL Diagnostic Manager. Each new license you add to SQL Diagnostic Manager adds to the total number of SQL Server instances that you can monitor.

To get a new license, contact your [IDERA Sales](#) representative.

Generate a new license key

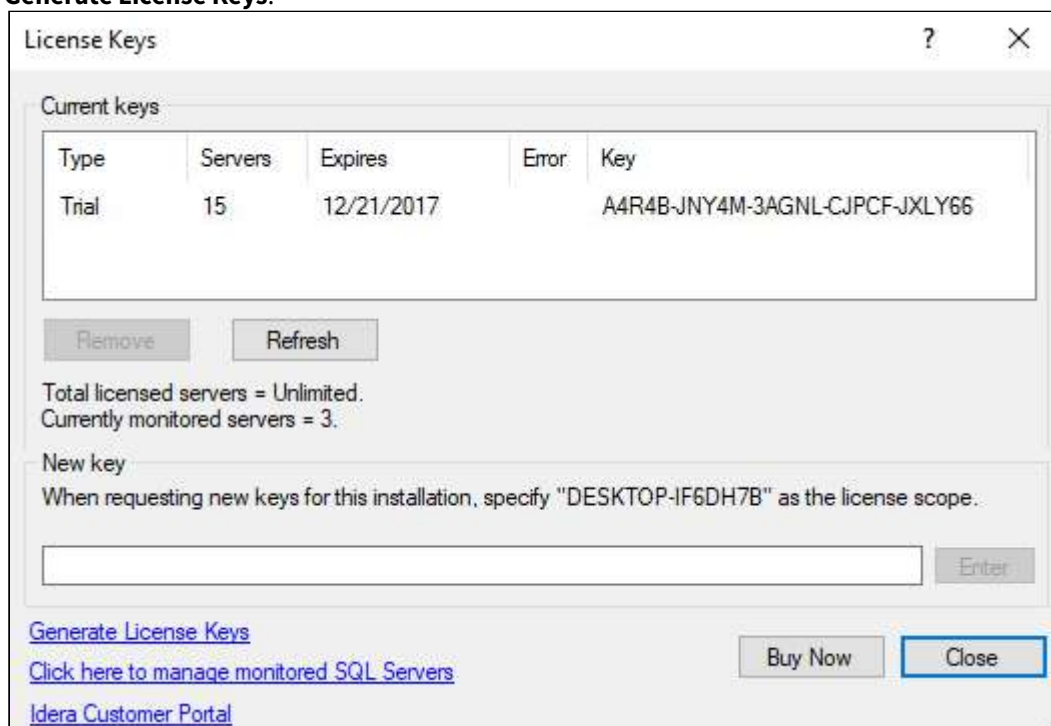
SQL Diagnostic Manager provides a new self-service licensing model that allows you to generate new license keys by logging in to the [IDERA Customer Portal](#) or by using the license manager utility.

⚠ To upgrade to SQL Diagnostic Manager 9.0 or higher versions, a new license key is required.

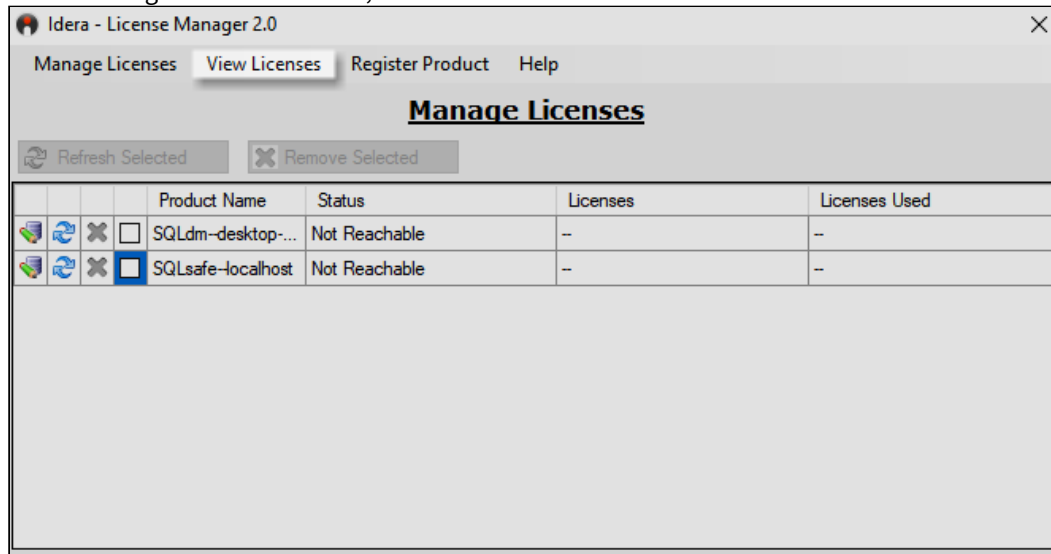
Log in the IDERA Customer portal and generate a new license key

To generate a new license key:

1. Open the License Keys window, click **?**, select **Manage Licenses** from the Toolbar menu, and click **Generate License Keys**.





- On the Manage Licenses window, click **View Licenses**.




- On the Login to the [IDERA Customer Portal](#) window, type your email address in the corresponding space.
- Enter your password in the appropriate space.
- Click **Log in** and follow the instructions in the [IDERA Customer Portal](#).

Use the license manager utility and generate a new license

To generate a new license key:

- Open the License Keys window, click , select **Manage Licenses** from the Toolbar menu, and click **Generate License Keys**.
- On the Manage Licenses window, click the Edit icon ().
- On the Enter credentials to connect window, type your username in the appropriate space.
- Enter your password in the corresponding space.
- Select the authentication method by clicking the **Windows** button or the **SQL** button.
- Click **Log in** and follow the instructions.

Manage your SQL Server instances

The Manage Servers window ( > **Manage Servers**) allows you to see all the SQL Server instances monitored by SQL Diagnostic Manager. You can:

- Add new SQL Server instances to SQL Diagnostic Manager
- Edit the SQL Diagnostic Manager configuration for a particular SQL Server instance
- Remove a SQL Server instance from SQL Diagnostic Manager
- Access the [VM Configuration window](#) to associate your monitored virtual SQL Server instances to a host machine
- Apply an existing alert template to a server or group of servers
- Test the connection status of your SQL Server instances

Edit your Management Service configuration

The SQLDM Management Service Configuration wizard allows you to change how the SQLDM Management Service interacts with the SQL Diagnostic Manager Repository.

You can edit the SQLDM Management Service settings whenever you change the location of your SQL Diagnostic Manager Repository or if you want to change the account or authentication method the SQLDM Management Service uses to connect to the SQL Diagnostic Manager Repository.

Access the Management Service Configuration wizard

You can access the SQLDM Management Service Configuration wizard by running the `SQLdmManagementServiceConsole.exe` file in your root SQL Diagnostic Manager directory.

To use the SQLdm Management Service Configuration wizard:

1. Run `SQLdmManagementServiceConsole.exe` in your root SQL Diagnostic Manager directory.
2. Click **Next** on the Welcome window.
3. Enter your Repository information.
4. Enter the Authentication method that you want the SQLdm Management Service to use to write configuration and collected statistics to the SQL Diagnostic Manager Repository.
5. Click **Test** to verify the connection.
6. Click **Next**.
7. Verify the changes are correct.
8. Click **Finish** to apply the changes.

Select a SQL Server instance for the Management Service

The SQLDM Management Service Configuration wizard allows you to select the SQL Server instance for the SQLDM Management Service to connect with the SQLDM Repository. Select the appropriate SQL Server instance from the lists of available local and network instances, and then click **OK**.

Select a database for the Management Service

The SQLDM Management Service Configuration wizard allows you to select the database for the SQLDM Management Service to connect with the SQLDM Repository. Select the appropriate database from the list of available databases, and then click **OK**.

Monitored metrics

SQL Diagnostic Manager collects a number of SQL Server and operating system performance metrics to help you monitor, alert, and report on your system health. SQL Diagnostic Manager includes a number of metrics for your use, such as:

- Batches Received (Per Second)
- Blocked Sessions (Count)
- Buffer Cache Hit Ratio (Percent)
- Client Computers
- Compiles (Per Second)
- Data Used (Percent)
- Database Status
- Disk Queue Length
- Disk Time (Percent)
- I/O Waits
- Lock Waits
- Log Flushes
- Log Space Used (Percent)
- Memory Page Faults (Per Second)
- Memory Paging
- Memory Waits
- Other Waits
- Packets Received (Per Second)
- Packets Sent and Received
- Packets Sent (Per Second)
- Page Reads (Per Second)
- Page Writes (Per Second)
- Percent of SQL Server Data Space Used
- Percent of SQL Server Log Space Used
- Physical Page Reads and Writes
- Processor Queue Length
- Re-compiles Per Second
- Signal Waits
- SQL Server Batches Received
- SQL Server CPU Usage
- SQL Server Log Space Used (Percent)
- SQL Server Memory Usage
- SQL Server Response Time
- Total Memory Usage
- Transaction Log Waits
- Transactions (Per Second)
- Wait type category - Backup
- Wait type category - I/O Page Latch
- Wait type category - Lock
- Wait type category - Memory
- Wait type category - Non-I/O Page Latch
- Wait type category - Non-Page Latch
- Wait type category - Other
- Wait type category - Transaction Log

Batches Received (Per Second)

The Batches Received (Per Second) metric counts the total number of batch requests that the SQL Server computer receives per second. This rate is a measurement of throughput on the processor.

This metric uses the counter `Batch requests/sec` from the SQL Server `SQL Statistics` counter object. On SQL Server 2000 instances, this counter is available through `sysperfinfo`. On SQL Server 2005 or later instances, this counter is available through `sys.dm_os_performance_counters`.

To track this metric, use the [CPU Call Rates chart](#) in the CPU dashboard of the Server Overview tab.

While this counter has no good or bad value, each of these T-SQL batches must be:

1. Transmitted to the server using the network.
2. Parsed by SQL Server to check syntax.
3. Have an execution plan compiled.
4. Executed according to the execution plan.
5. Have the results transmitted back to the client computer or Web site.

Reduce the number and impact of T-SQL batches

In this situation, you should reduce the number of T-SQL batches and the impact of each as much as possible by:

- Reducing the size and complexity of the T-SQL by creating a stored procedure containing the commands you want executed and then simply calling it in the T-SQL batch. This approach reduces the size and complexity of the T-SQL batch, which means less network traffic and less CPU time consumed producing a complex execution plan.
- Once you move most of the T-SQL code into stored procedures, consider grouping as many of those calling T-SQL commands into a single network batch, if possible. Also consider redesigning the application to call only those stored procedures you need to complete an application function all from within one batch rather than sending each EXECUTE statement to execute separately in its own batch. This strategy reduces network queues dramatically as better use is made of the default 4 KB network packet size.

Blocked Sessions (Count)

The Blocked Sessions (Count) metric counts the number of sessions on the SQL Server instance blocked by other sessions holding requested locks.

By default, a blocked session waits forever for a lock. You can change this behavior by adding `SET LOCKTIMEOUT 60000` to the top of the first T-SQL batch after you make a connection.

Buffer Cache Hit Ratio (Percent)

The Buffer Cache Hit Ratio (Percent) metric calculates the percentage of data pages SQL Server found in memory without reading the data from the disk subsystem. This value is measured by reading the 'buffer cache hit ratio' from `sysperfinfo`. The higher this value is the more efficiently the system is running.

Reduce the percentage of memory used by SQL Server

Once SQL Server is up for at least 30 minutes, use the following solutions if values lower than 85% regularly occur:

- Add more physical memory (RAM) to the computer.
- If your site makes use of extended stored procedures that are infrequently called, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never again called, which then wastes available memory. A `DBCC dllname (FREE)` releases that memory for use by both the procedure cache and Data Pages, which has a significant positive effect on both the Procedure Cache Hit Ratio and the Buffer Cache Hit Ratio. Execute `sp_helpextendedproc` to view the extended stored procedures currently loaded in memory by SQL Server.
- Allowing SQL Server to consume more of the available memory, making sure that OS Paging does not increase.
- Limit SQL Server computers to performing only SQL Server work. Stop any unnecessary programs, such as allowing the computer to act as either a primary or backup domain controller.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.

Client Computers

The Client Computers metric calculates the number of distinct connected network adapters currently using the SQL Server instance. This value is measured by getting the count of processes from distinct computers in `sysprocesses` table.

SQL Compilations (Per Second)

This metric calculates the number of SQL compilations per second, including statement-level re-compilations.

This metric uses the counter `SQL Compilations/sec` from the SQL Server `SQL Statistics` counter object. On SQL Server 2000 instances, this counter is available through `sysperfinfo`. On SQL Server 2005 or later instances, this counter is available through `sys.dm_os_performance_counters`.

A high number of compilations (greater than 100 per second) can indicate a high server workload or that there is more recompilation overhead than necessary. Too many compilations can impact a server's performance.

Data Used (Percent)

The Data Used (Percent) metric calculates the size of all data files used as a percent of the added allocated size of all data files.

In a typical medium-to-large-scale OLTP environment with a data file size of over 50 GB, this value slowly increases over time in a linear fashion with the population of any available history tables.

Reduce the percentage of data used

If your site experiences a sudden sharp increase in this value of over 10% in an hour, investigate the issue to fully understand the underlying causes which often lead to poor performance. Possible causes include:

- A large number of rows were inserted into a table. If so, then make sure that the table is not fragmented due to this operation. To verify, run `DBCC SHOWCONTIG`.
- An index with very poor cardinality (uniqueness) was added to a large table. Indexes with poor cardinality often exceed the size of the underlying table itself and in almost all cases are not used by any SQL Server query.
- A large number of rows were incorrectly changed by performing an insert of a new row followed by a delete of the old row. Although the net effect of the change is essentially the same number of rows; the database would have grown significantly in size and require you to issue a `DBCC SHRINKDATABASE`. It is likely that the table is also fragmented because the inserts were done before the deletes so run `DBCC SHOWCONTIG`. Always delete the old rows out of a table first before inserting the new rows; otherwise the tables are fragmented more quickly, databases increase in size, and the performance of the transactions is degraded due to the large number of page splits the table had to undergo.

Database Status

The Database Status metric indicates the operational status of the database, such as Normal, Offline, or Suspect. You can select the database status for which you want to receive alerts.

Disk Queue Length

The Disk Queue Length metric counts the average number of system requests that are waiting for disk access on the computer hosting the SQL Server instance. The Disk view on the Resources tab allows you to track your Disk Queue Length, along with other key disk statistics, over a period of time on the computer hosting your SQL Server instance.

Reasons why this metric has no value

SQL Diagnostic Manager is unable to collect the corresponding counter due to one of the following reasons:

- It usually takes two refreshes to begin populating the Disk Queue metric, which occurs within a few seconds.
- OS Metrics collection is disabled or not working properly. You must enable OS Metrics collection for SQLdm to collect hardware performance metrics.
- Lightweight Pooling is enabled. SQL Diagnostic Manager cannot collect OS metrics if Lightweight Pooling is enabled on the SQL Server instance.
- The counter does not exist on this SQL Server instance.

Disk Time (Percent)

The Disk Time (Percent) metric calculates the percentage of elapsed time during which all the disks were busy servicing read and write requests on the SQL Server computer. This counter is measured by subtracting the Disk Idle Time percent from 100. The WMI object `Win32_PerfRawData_PerfDisk_PhysicalDisk` property `PercentIdleTime` is measured to get Disk Idle Time percent.

Reduce disk time

If this metric value regularly exceeds 75%, consider the following solutions:

- Check OS Paging to make sure that paging from/to the swap file is not causing these IOs.
- Distribute the data and logs over more disks.
- Replace disks with faster disks.
- Add more memory if the main cause is Disk Reads.
- Switch from a RAID 5 to RAID 10 solution if Disk Writes make up more than 20% of total disk access.
- Move non-SQL Server applications to another computer.
- If the computer is running multiple instances of SQL Server, consider placing each instance on a separate computer.

I/O Waits

The I/O Waits metric calculates the length of time that SQL Server processes had to wait for I/O to become available. This rate is a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload.

Lock Waits

The Lock Waits metric calculates the length of time that SQL Server processes had to wait for a resource lock to resolve. This rate is a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload.

Log Flushes (Per Second)

The Log Flushes (Per Second) metric reads LOG_FLUSHES/SEC from `sysperfinfo`. Each time data changes, the changes write to the log cache, which resides in memory. Under certain conditions, this log cache is flushed to disk. These situations include:

- A commit or roll back of an explicit or implicit transaction.
- A CHECKPOINT statement execution.
- SQL Server issues an automatic checkpoint.
- The Log Cache is full and needs emptying.

Reduce log flushes

On an OLTP system that runs relatively small transactions (on average, less than 40 rows added, changed, or deleted per transaction), the number of Log Flushes should not exceed SQL Batches by more than 40% during periods of high activity. If you exceed this percentage, then one of the following abnormal conditions exists:

- T-SQL batches sent for execution contain a number of individual data manipulation statements (inserts, deletes, or updates) not enclosed within a transaction.
- Large batch jobs which update significant amounts of data are run during the day.
- Deadlocks or Lock Timeouts are causing a number of transactions to roll back or execute a ROLLBACK TRANSACTION statement.
- Your Recovery Interval parameter is too low. Raise this parameter to 10-15 minutes.
- An explicit CHECKPOINT statement is executed within some transactions.
- If you need to delete the entire contents of a table, use the TRUNCATE TABLE statement instead of DELETE as it produces less log flushes.

If you implement the previously-mentioned measures and the Log Flushes still greatly exceed 140% of SQL Batches, then consider:

- Switching your RAID array from a RAID 5 to a RAID 10 solution. Each write IO results in two writes using RAID 10 vs. 4 for RAID 5 (100% more efficient with writes).
- As long as the RAID controller has some form of battery backup, then switch its cache mode from Write-through to Write-back, as this setting increases the system's ability to handle write IOs by an order of magnitude.
- Adding more cache memory to the RAID controller.

Memory Page Faults (Per Second)

This metric calculates the page fault rate for all processes on the SQL Server computer. The WMI object `Win32_PerfRawData_PerfOS_Memory` property `PagesPersec` is measured to get this value. For optimal performance, the page fault rate should either be either 0 or very low.

Reduce memory page faults

Use the following solutions to reduce the page fault rate if it rises above 100 for a period of time longer than a momentary spike:

- Decrease the memory being used by stopping unnecessary programs such as the Print Spooler from running on the dedicated SQL Server computer. You should also eliminate activities such as the server acting as a PDC or BDC and (as a last resort) reducing the amount of memory SQL Server may consume.
- Add more physical memory on the computer.
- If the computer is running multiple instances of SQL Server, then consider moving each instance to a separate physical computer.

Memory Paging

The Memory Paging metric counts the number of pages being swapped in and out of memory per second. The **Memory** view on the **Resources** tab allows you to track paging and other key memory statistics over a period of time.

Reasons why this metric has no value

SQL Diagnostic Manager is unable to collect the corresponding counter due to one of the following reasons:

- It usually takes two refreshes to begin populating the Memory Paging metric, which occurs within a few seconds.
- OS Metrics collection is disabled or not working properly. You must enable OS Metrics collection for SQLdm to collect hardware performance metrics.
- Lightweight Pooling is enabled. SQL Diagnostic Manager cannot collect OS metrics if Lightweight Pooling is enabled on the SQL Server instance.
- The counter does not exist on this SQL Server instance.

Memory Waits

This metric calculates the length of time that SQL Server processes had to wait for SQL Server memory to become available. This rate is a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload.

Other Waits

This metric includes any additional waits occurring on this SQL Server outside of the following wait categories:

- [I/O](#)
- [Locks](#)
- [Memory](#)
- [Signal](#)

Waits are a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload. To get more details about monitored wait statistics, see the [Server Waits view](#), and review the Microsoft Best Practice Article [SQL Server Waits and Queues](#).

Packets Received (Per Second)

This metric is calculated as a rate between two refreshes of the value of the system statistical function @@pack_received. If this value is consistently high, consider changing the packet size.

Packets Sent and Received

This metric counts the number of packets being sent and received, per second, on your SQL Server instance.

Packets Sent (Per Second)

This metric is calculated as a rate between two refreshes of the value of the system statistical function @@pack_sent. If this value is consistently high, consider changing the packet size.

Page Reads (Per Second)

This metric counts the number of physical reads the disk sub-system performs to bring SQL Server database pages into the data (buffer) cache. For the first few minutes after you start or restart SQL Server, almost any request for data is retrieved from disk. As the data cache fills, these requests are reduced. This value is measured by reading 'PAGE_READS/SEC' from `sysperfinfo`.

Currently, even the fastest disks can support a maximum sustained rate of around only 85 random IOs per second. To determine if your IO subsystem is "flooded," perform the following calculation:

On RAID 10: $((\text{Page reads} + \text{read ahead pages} + ((\text{page writes} + \text{checkpoint pages} + \text{lazy writes} + \text{log flushes}) \times 2)) / \text{number of disk spindles in RAID array})$ must be < 65 .

Reduce page reads

If the page reads rate exceeds 65 for more than five seconds every 20 minutes, then consider these solutions:

- Add more physical memory (RAM) to the computer.
- If your site makes use of extended stored procedures that are not called very often, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never called again. This is tremendously wasteful in terms of available memory and a `DBCC DLLNAME (FREE)` releases that memory for use by both the procedure cache as well as the Data Pages which have a significant positive effect on both the Procedure Cache Hit Ratio as well as the Buffer Cache Hit Ratio. In order to view the extended stored procedures currently loaded in memory by SQL Server, execute `SP_HELPXTEENDEDPROC`.
- Allow SQL Server to consume more of the available memory (making sure that OS Paging does not increase).
- If you are running large, batch-style reporting queries on the OLTP server, then large numbers of data pages are flushed from cache causing much additional IO overhead. In this situation, set up a separate SQL Server Reports computer to service all but the smallest reports. You can feed this Reports Server via backups and Log Shipping from the OLTP server.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.

Page Writes (Per Second)

This metric counts the number of physical database writes the disk sub-system performs. This value is measured by reading `PAGE_READS/SEC` from `sysperfinfo`.

Currently, even the fastest disks can support a maximum sustained rate of around only 85 random IOs per second. To determine if your IO subsystem is "flooded," perform the following calculation:

- On RAID 5: $((\text{Page reads} + \text{read ahead pages} + ((\text{page writes} + \text{checkpoint pages} + \text{lazy writes} + \text{log flushes}) \times 4)) / \text{number of disk spindles in RAID array})$ must be < 65 .
- On RAID 10: $((\text{Page reads} + \text{read ahead pages} + ((\text{page writes} + \text{checkpoint pages} + \text{lazy writes} + \text{log flushes}) \times 2)) / \text{number of disk spindles in RAID array})$ must be < 65 .

Reduce page writes

If the page writes rate exceeds 65 for more than five seconds every 20 minutes, then consider the following:

- Add more physical memory (RAM) to the computer.
- If your site makes use of extended stored procedures that are not called very often, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never called again! This is tremendously wasteful in terms of available memory and a `DBCC DLLNAME (FREE)` releases that memory for use by both the procedure cache as well as the Data Pages which has a significant positive effect on both the Procedure Cache Hit Ratio as well as the Buffer Cache Hit Ratio. In order to view the extended stored procedures currently loaded in memory by SQL Server, execute `SP_HELPXTEENDEDPROC`.
- Allow SQL Server to consume more of the available memory (making sure that OS Paging does not increase).
- If you are running large, batch-style reporting queries on the OLTP server, then large numbers of data pages are flushed from cache causing much additional IO overhead. In this situation, set up a separate SQL Server Reports computer to service all but the smallest reports. You can feed this Reports Server via backups and Log Shipping from the OLTP server.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.

Percent of SQL Server Data Space Used

This metric calculates the percentage of the allocated disk space for databases that is being used by the SQL Server instance. The total size allocated to the databases is the sum of unused space, data space, and index space. Data space includes leaf level clustered index pages, since these pages are the data pages. Index space excludes the leaf pages of clustered indexes. The Files view on the Databases tab allows you to view detailed data file information including the data size and the percentage of allocated file size being used on each of your databases.

Percent of SQL Server Log Space Used

This metric calculates the percentage of the allocated space for logs that is currently being used. The Files view on the Databases tab allows you to view detailed log information including the log size and the percentage of allocated file size being used on each of your databases.

Physical Page Reads and Writes

This metric counts the number of physical page reads and writes to disk from the SQL Server instance.

Processor Queue Length

This metric counts the number of ready threads in the processor queue on the computer that hosts the SQL Server instance. The WMI object `Win32_PerfRawData_PerfOS_System` property `ProcessorQueueLength` is measured to get processor queue length count.

Re-compiles (Per Second)

The Re-compiles (Per Second) metric calculates the rate of re-compiles on your monitored instance, using the counter SQL Re-Compilations/sec from the SQL Server SQL Statistics counter object.

On SQL Server 2000 instances, this counter is available through `sysperfinfo` and represents the rate of batch re-compilations per second.

On SQL Server 2005 or later instances, this counter is available through `sys.dm_os_performance_counters` and represents the rate of statement-level re-compilations per second.

Typically, this metric value is low. A high re-compilation rate may be a sign of excessive re-compilation overhead. Causes of re-compiles include various events including schema changes, statistics updates, and explicit calls to `WITH RECOMPILE`. Excessive re-compilation impacts your server performance.

Signal waits

This metric calculates the length of time that SQL Server processes had to wait for CPU time after the associated resources became available for processing. This rate is a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload.

SQL Server Batches Received (Per Second)

The SQL Server Batches Received (Per Second) metric counts the number of T-SQL batches that SQL Server receives for execution every second. This value is measured by reading 'batch requests/sec' from `sysperfinfo`. While this counter has no good or bad value, each of these T-SQL batches must:

1. Be transmitted to the server using the network.
2. Be parsed by SQL Server to check syntax.
3. Include a compiled execution plan.
4. Include an executed execution plan.
5. Transmit the results back to the client computer or Web.

You must minimize the number and reduce the impact of these T-SQL batches as much as possible by:

- Reducing the size and complexity of the T-SQL by creating a stored procedure containing the commands you want executed, and then simply calling it in the T-SQL batch. This action reduces the size and complexity of the T-SQL batch, resulting in less network traffic and CPU time consumed producing a complex execution plan.
- Grouping as many calling T-SQL commands into a single network batch after you move most of the T-SQL code into stored procedures. Redesign the application to call whatever stored procedures you need to complete an application function all from within one batch rather than sending each EXECUTE statement to execute separately in its own batch. This action dramatically reduces network queues as you make better use of the default network packet size of 4 KB.

SQL Server CPU Usage (Percent)

The SQL Server CPU Usage (Percent) metric calculates the percentage of CPU time used by the SQL Server instance. This value is calculated by the percentage of time in a query spent by the I/O, idle, and CPU times on the computer that hosts the SQL Server instance.

To track this metric, use the [CPU Usage chart](#) in the CPU dashboard of the Server Overview tab.

Lower SQL Server CPU usage

If this value regularly exceeds 75%, consider taking the following actions:

- Reduce the number of SQL re-compilations, as they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled. You can remove most of these reasons by careful coding.
- Make sure that all T-SQL statements (whether in a Stored Procedure, Trigger, or Ad Hoc statement) that reference objects fully qualify the object referenced.

For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. Both of these situations consume a significant amount of CPU time.

Log Space Used (Percent)

The Log Space Used (Percent) metric calculates the added size all log files used as a percent of summed allocated size of all data files.

A transaction log continues to grow until its inactive portion is truncated. Transaction log truncation occurs because either:

- the BACKUP LOG statement successfully concluded, or
- if the database is using the Simple Recovery Model, then a CHECKPOINT T-SQL statement executes or SQL Server performed an internal automatic CHECKPOINT. SQL Server performs this action if the log reaches 70% full or when SQL Server determines the length of time to recover from the log, in the event of a failure, exceeds the Recovery Interval option.

Only the inactive portion of the log is ever removed. The inactive portion of the log is the portion of the log up to but not including the start date/time of the oldest open transaction. No matter which method is used to truncate the log, if there is a perpetually open transaction then the log continues to grow in size until it consumes all available disk space.

Reduce the percentage of the log used

If your transaction log size continues to grow despite issuing BACKUP LOG or CHECKPOINT statements, check for a rogue transaction. Begin your research by sorting the Last Batch column on the SQL Processes grid into ascending mode such that the processes containing the oldest batches appear first. Scroll down the list until you find the first user process with a non-zero number of transactions in its Trans column, and then kill this process. Once this is done, perform a CHECKPOINT or BACKUP LOG to reduce the Log File Used Size.

SQL Server Memory Usage

This metric calculates the percentage total memory being used on the computer hosting the selected instance by SQL Server. The Memory view on the Resources tab allows you to track the SQL Server memory usage, along with other key memory statistics, on the computer hosting your SQL Server instance.

SQL Server Response Time (ms)

This metric calculates the time (in milliseconds) that SQL Diagnostic Manager currently needs to send a simple SQL command to the SQL Server instance, have it processed, and receive the returned result set.

Typically, this value should average below 2000 ms for a well tuned network and server. Average values in excess of 2000 ms indicate either an excessively busy network segment or a stressed SQL Server.

Total Memory Usage

This metric calculates the percentage of physical memory and virtual memory Windows is using. The Memory view on the Resources tab allows you to track the Total Memory usage along with other key memory statistics on the computer hosting your SQL Server instance.

Transaction Log Waits

This metric calculates the length of time that SQL Server processes had to wait for the transaction log to become available. This rate is a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload.

Transactions (Per Second)

This metric calculates the total number of transactions per second across all databases on the monitored instance.

This metric uses the counter `Transactions/sec` from the SQL Server Databases counter object and `_Total` instance. On SQL Server 2000 instances, this counter is available through `sysperfinfo`. On SQL Server 2005 or later instances, this counter is available through `sys.dm_os_performance_counters`.

This rate is a measurement of throughput on the processor, as determined by the SQL Server workload. Although this rate measures activity within a transaction, and a higher transaction rate indicates a higher risk of resource issues, it does not account for all activity on the SQL Server computer.

Wait type category - Backup

The Backup wait type category includes all the waits that are associated with bottlenecks caused during a backup process, including the following:

- BACKUP
- BACKUP_CLIENTLOCK
- BACKUP_OPERATOR
- BACKUPBUFFER
- BACKUPIO
- BACKUPTHREAD
- DISKIO_SUSPEND

Wait type category - I/O Page Latch

The I/O Page Latch wait type category includes all the waits that are associated with non-I/O page latch bottlenecks, including the following:

- ASYNC_IO_COMPLETION
- IO_COMPLETION
- PAGEIOLATCH_DT
- PAGEIOLATCH_EX
- PAGEIOLATCH_KP
- PAGEIOLATCH_NL
- PAGEIOLATCH_SH
- PAGEIOLATCH_UP

Wait type category - Lock

The Lock wait type category includes all the waits that are caused when a task is waiting to acquire a lock, including the following:

- LCK_M_BU
- LCK_M_IS
- LCK_M_IU
- LCK_M_IX
- LCK_M_RIn_NL
- LCK_M_RIn_S
- LCK_M_RIn_U
- LCK_M_RIn_X
- LCK_M_RS_S
- LCK_M_RS_U
- LCK_M_RX_S
- LCK_M_RX_U
- LCK_M_RX_X
- LCK_M_S
- LCK_M_SCH_M
- LCK_M_SCH_S
- LCK_M_SIU
- LCK_M_SIX
- LCK_M_U
- LCK_M_UIX
- LCK_M_X
- XACTLOCKINFO

Wait type category - Memory

The Memory wait type category includes all the waits that are associated with memory bottlenecks, including the following:

- CMEMTHREAD
- RESOURCE_SEMAPHORE
- RESOURCE_SEMAPHORE_MUTEX
- RESOURCE_SEMAPHORE_QUERY_COMPILE
- RESOURCE_SEMAPHORE_SMALL_QUERY
- SOS_RESERVEDMEMBLOCKLIST

Wait type category - Non-I/O Page Latch

The Non-I/O Page Latch wait type category includes all the waits that are associated with non-I/O page latch bottlenecks, including the following:

- PAGELATCH_DT
- PAGELATCH_EX
- PAGELATCH_KP
- PAGELATCH_NL
- PAGELATCH_SH
- PAGELATCH_UP

Wait type category - Non-Page Latch

The Non-Page Latch wait type category includes all the waits that are caused when a task is waiting for a latch that is not related to paging, including the following:

- LATCH_DT
- LATCH_EX
- LATCH_KP
- LATCH_NL
- LATCH_SH
- LATCH_UP
- TRAN_MARKLATCH_DT
- TRAN_MARKLATCH_EX
- TRAN_MARKLATCH_KP
- TRAN_MARKLATCH_NL
- TRAN_MARKLATCH_SH
- TRAN_MARKLATCH_UP

Wait type category - Other

This wait type category includes all the waits that do not belong to the Memory, I/O, Lock, Transaction Log, Non-I/O Page Latch, Non-Page Latch, or Backup categories.

The following waits are associated with the Other category:

- ABR
- ASSEMBLY_LOAD
- ASYNC_DISKPOOL_LOCK
- ASYNC_NETWORK_IO
- AUDIT_GROUPCACHE_LOCK
- AUDIT_LOGINCACHE_LOCK
- AUDIT_ON_DEMAND_TARGET_LOCK
- AUDIT_XE_SESSION_MGR
- BROKER_CONNECTION_TASK
- BROKER_ENDPOINT_STATE_TASK
- BROKER_INIT
- BROKER_MASTERSTART
- BROKER_REGISTERALLENDPOINTS
- BROKER_SERVICE
- BROKER_SHUTDOWN
- BROKER_XMIT_SHUTDOWN
- BROKER_XMIT_START
- BROKER_XMIT_TABLE
- BUILTIN_HASHKEY_MUTEX
- CHECK_PRINT_RECORD
- CLEAR_DB
- CLR_AUTO_EVENT
- CLR_CRST
- CLR_JOIN
- CLR_MANUAL_EVENT
- CLR_MEMORY_SPY
- CLR_MONITOR
- CLR_RWLOCK_READER
- CLR_RWLOCK_WRITER
- CLR_TASK_START
- CLRHOST_STATE_ACCESS
- COMMIT_TABLE
- CURSOR
- CURSOR_ASYNC
- CXPACKET
- CXROWSET_SYNC
- DAC_INIT
- DBMIRROR_DBM_EVENT
- DBMIRROR_DBM_MUTEX
- DBMIRROR_SEND
- DBMIRROR_WORKER_QUEUE
- DBMIRRORING_CMD
- DBTABLE
- DEADLOCK_ENUM_MUTEX
- DEADLOCK_TASK_SEARCH
- DEBUG

- DIRTY_PAGE_POLL
- DISABLE_VERSIONING
- DISPATCHER_QUEUE_SEMAPHORE
- DLL_LOADING_MUTEX
- DROPTEMP
- DTC
- DTC_ABORT_REQUEST
- DTC_RESOLVE
- DTC_STATE
- DTC_TMDOWN_REQUEST
- DTC_WAITFOR_OUTCOME
- DUMP_LOG_COORDINATOR
- DUMP_LOG_COORDINATOR_QUEUE
- DUMPTRIGGER
- EC
- EE_PMOLOCK
- EE_SPECPROC_MAP_INIT
- ENABLE_VERSIONING
- ERROR_REPORTING_MANAGER
- EXCHANGE
- EXESYNC
- EXECUTION_PIPE_EVENT_INTERNAL
- FAILPOINT
- FCB_REPLICA_READ
- FCB_REPLICA_START
- FCB_REPLICA_WRITE
- FS_FC_RWLOCK
- FS_GARBAGE_COLLECTOR_SHUTDOWN
- FS_HEADER_RWLOCK
- FS_LOGTRUNC_RWLOCK
- FSA_FORCE_OWN_XACT
- FSAGENT
- FSTR_CONFIG_MUTEX
- FSTR_CONFIG_RWLOCK
- FT_COMPROWSET_RWLOCK
- FT_IPTS_RWLOCK
- FT_IPTSISM_MUTEX
- FT_MASTER_MERGE
- FT_METADATA_MUTEX
- FT_RESTART_CRAWL
- FT_RESUME_CRAWL
- FULLTEXT GATHERER
- GUARDIAN
- HADR_FILESTREAM_IOMGR_IO_COMPLETION
- HTTP_ENDPOINT_COLLCREATE
- HTTP_ENUMERATION
- HTTP_START
- IMP_IMPORT_MUTEX
- IMPPROV_IOWAIT
- INDEX_USAGE_STATS_MUTEX
- INTERNAL_TESTING
- IO_AUDIT_MUTEX
- IO_RETRY

- IOAFF_RANGE_QUEUE
- KTM_ENLISTMENT
- KTM_RECOVERY_MANAGER
- KTM_RECOVERY_RESOLUTION
- LOGGENERATION
- LOGMGR_FLUSH
- [LOGMGR_RESERVE_APPEND](#)
- LOWFAIL_MEMMGR_QUEUE
- METADATA_LAZYCACHE_RWLOCK
- MIRROR_SEND_MESSAGE
- MSQL_DQ
- MSQL_SYNC_PIPE
- MSQL_XACT_MGR_MUTEX
- MSQL_XACT_MUTEX
- MSQL_XP
- MSSEARCH
- NET_WAITFOR_PACKET
- NODE_CACHE_MUTEX
- OLEDB
- PARALLEL_BACKUP_QUEUE
- PARALLEL_REDO_WORKER_SYNC
- PARALLEL_REDO_WORKER_WAIT_WORK
- PERFORMANCE_COUNTERS_RWLOCK
- PLACEHOLDER1
- PLACEHOLDER2
- PREEMPTIVE_ABR
- PREEMPTIVE_AUDIT_ACCESS_EVENTLOG
- PREEMPTIVE_AUDIT_ACCESS_SECLOG
- PREEMPTIVE_CLOSEBACKUPMEDIA
- PREEMPTIVE_CLOSEBACKUPTAPE
- PREEMPTIVE_CLOSEBACKUPVDIDEVICE
- PREEMPTIVE_CLUSAPI_CLUSTERRESOURCECONTROL
- PREEMPTIVE_COM_COCREATEINSTANCE
- PREEMPTIVE_COM_COGETCLASSOBJECT
- PREEMPTIVE_COM_CREATEACCESSOR
- PREEMPTIVE_COM_DELETEROWS
- PREEMPTIVE_COM_GETCOMMANDTEXT
- PREEMPTIVE_COM_GETDATA
- PREEMPTIVE_COM_GETNEXTROWS
- PREEMPTIVE_COM_GETRESULTS
- PREEMPTIVE_COM_GETROWSBYBOOKMARK
- PREEMPTIVE_COM_LBFLUSH
- PREEMPTIVE_COM_LBLOCKREGION
- PREEMPTIVE_COM_LBREADAT
- PREEMPTIVE_COM_LBSETSIZE
- PREEMPTIVE_COM_LBSTAT
- PREEMPTIVE_COM_LBUNLOCKREGION
- PREEMPTIVE_COM_LBREADAT
- PREEMPTIVE_COM_LBSTAT
- PREEMPTIVE_COM_LBUNLOCKREGION
- PREEMPTIVE_COM_LBWRITEAT
- PREEMPTIVE_COM_QUERYINTERFACE
- PREEMPTIVE_COM_RELEASE

- PREEMPTIVE_COM_RELEASEACCESSOR
- PREEMPTIVE_COM_RELEASEROWS
- PREEMPTIVE_COM_RELEASESESSION
- PREEMPTIVE_COM_RESTARTPOSITION
- PREEMPTIVE_COM_SEQSTRMREAD
- PREEMPTIVE_COM_SESSTRMREADANDWRITE
- PREEMPTIVE_COM_SETDATAFAILURE
- PREEMPTIVE_COM_SETPARAMETERINFO
- PREEMPTIVE_COM_SETPARAMETERPROPERTIES
- PREEMPTIVE_COM_STRMLOCKREGION
- PREEMPTIVE_COM_STRMSEEKANDREAD
- PREEMPTIVE_COM_STRMSEEKANDWRITE
- PREEMPTIVE_COM_SRTMSETSIZE
- PREEMPTIVE_COM_SRTMSTAT
- PREEMPTIVE_COM_STRMUNLOCKREGION
- PREEMPTIVE_CONSOLEWRITE
- PREEMPTIVE_CREATEPARAM
- PREEMPTIVE_DEBUG
- PREEMPTIVE_DFADDLINK
- PREEMPTIVE_DFSLINKEXISTCHECK
- PREEMPTIVE_DFSLINKHEALTHCHECK
- PREEMPTIVE_DFSREMOVELINK
- PREEMPTIVE_DFSREMOVEROOT
- PREEMPTIVE_DFSROOTFOLDERCHECK
- PREEMPTIVE_DFSROOTINIT
- PREEMPTIVE_DFSROOTSHARECHECK
- PREEMPTIVE_DTV_ABORT
- PREEMPTIVE_DTC_ABORTREQUESTDONE
- PREEMPTIVE_DTC_BEGINTRANSACTION
- PREEMPTIVE_DTC_COMMITREQUESTDONE
- PREEMPTIVE_DTC_ENLIST
- PREEMPTIVE_DTC_PREPAREREQUESTDONE
- PREEMPTIVE_FILESIZEGET
- PREEMPTIVE_FSAOLEDB_ABORTTRANSACTION
- PREEMPTIVE_FSAOLEDB_COMMITTRANSACTION
- PREEMPTIVE_FSRECOVER_UNCONDITIONALUNDO
- PREEMPTIVE_GETRMINFO
- PREEMPTIVE_LOCKMONITOR
- PREEMPTIVE_MSS_RELEASE
- PREEMPTIVE_ODBCOPS
- PREEMPTIVE_OLE_UNINIT
- PREEMPTIVE_OLEDB_ABORTORCOMMITTRAN
- PREEMPTIVE_OLEDB_ABORTTRAN
- PREEMPTIVE_OLEDB_GETDATASOURCE
- PREEMPTIVE_OLEDB_GETLITERALINFO
- PREEMPTIVE_OLEDB_GETPROPERTIES
- PREEMPTIVE_OLEDB_GETPROPERTYINFO
- PREEMPTIVE_OLEDB_GETSCHEMALOCK
- PREEMPTIVE_OLEDB_JOINTRANSACTION
- PREEMPTIVE_OLEDB_RELEASE
- PREEMPTIVE_OLEDB_SETPROPERTIES
- PREEMPTIVE_OLEDBBOPS
- PREEMPTIVE_OS_ACCEPTSECURITYCONTEXT

- PREEMPTIVE_OS_ACQUIRECREDENTIALSHANDLE
- PREEMPTIVE_OS_AUTHENTICATIONOPS
- PREEMPTIVE_OS_AUTHORIZATIONOPS
- PREEMPTIVE_OS_AUTHZGETINFORMATIONFROMCONTEXT
- PREEMPTIVE_OS_AUTHZINITIALIZECONTEXTFROMSID
- PREEMPTIVE_OS_AUTHZINITIALIZERESOURCEMANAGER
- PREEMPTIVE_OS_BACKUPREAD
- PREEMPTIVE_OS_CLOSEHANDLE
- PREEMPTIVE_OS_CLUSTEROPS
- PREEMPTIVE_OS_COMOMOPS
- PREEMPTIVE_OS_COMPLETEAUTHTOKEN
- PREEMPTIVE_OS_COPYFILE
- PREEMPTIVE_OS_CREATEDIRECTORY
- PREEMPTIVE_OS_CREATEFILE
- PREEMPTIVE_OS_CRYPTACQUIRECONTEXT
- PREEMPTIVE_OS_CRYPTIMPORTKEY
- PREEMPTIVE_OS_CRYPTOPS
- PREEMPTIVE_OS_DECRYPTMESSAGE
- PREEMPTIVE_OS_DELETEFILE
- PREEMPTIVE_OS_DELETESECURITYCONTEXT
- PREEMPTIVE_OS_DEVICEOPS
- PREEMPTIVE_OS_DIRSVC_NETWORKOPS
- PREEMPTIVE_OS_DISCONNECTNAMEDPIPE
- PREEMPTIVE_OS_DOMAINSERVICEOPS
- PREEMPTIVE_OS_DSGETDCNAME
- PREEMPTIVE_OS_DTCOPS
- PREEMPTIVE_OS_ENCRYPTMESSAGE
- PREEMPTIVE_OS_FILEOPS
- PREEMPTIVE_OS_FINDFILE
- PREEMPTIVE_OS_FLUSHFILEBUFFERS
- PREEMPTIVE_OS_FORMATMESSAGE
- PREEMPTIVE_OS_FREECREDENTIALSHANDLE
- PREEMPTIVE_OS_FREELIBRARY
- PREEMPTIVE_OS_GENERICOPS
- PREEMPTIVE_OS_GETADDRINFO
- PREEMPTIVE_OS_COMPRESSEDFILESIZE
- PREEMPTIVE_OS_GETDISKFREESPACE
- PREEMPTIVE_OS_GETDILEATTRIBUTES
- PREEMPTIVE_OS_GETFILESIZE
- PREEMPTIVE_OS_GETLONGPATHNAME
- PREEMPTIVE_OS_GETPROCADDRESS
- PREEMPTIVE_OS_GETVOLUMENAMEFORBOLUMEMOUNTPOINT
- PREEMPTIVE_OS_GETVOLUMEPATHNAME
- PREEMPTIVE_OS_INITIALIZESECURITYCONTEXT
- PREEMPTIVE_OS_LIBRARYOPS
- PREEMPTIVE_OS_LOGONUSER
- PREEMPTIVE_OS_LOOKUPACCOUNTSID
- PREEMPTIVE_OS_MESSAGEQUEUEOPS
- PREEMPTIVE_OS_MOVEFILE
- PREEMPTIVE_OS_NETGROUPGETUSERS
- PREEMPTIVE_OS_NETLOCALGROUPGETMEMBERS
- PREEMPTIVE_OS_NETUSERGETGROUPS
- PREEMPTIVE_OS_NETUSERGETLOCALGROUPS

- PREEMPTIVE_OS_NETUSERMODALSGET
- PREEMPTIVE_OS_NETVALIDATEPASSWORDPOLICY
- PREEMPTIVE_OS_NETVALIDATEPASSWORDPOLICYFREE
- PREEMPTIVE_OS_OPENDIRECTORY
- PREEMPTIVE_OS_PIPEOPS
- PREEMPTIVE_OS_PROCESSOPS
- PREEMPTIVE_OS_QUERYCONTEXTATTRIBUTES
- PREEMPTIVE_OS_QUERYREGISTRY
- PREEMPTIVE_OS_QUERYSECURITYCONTEXTTOKEN
- PREEMPTIVE_OS_REMOVEDIRECTORY
- PREEMPTIVE_OS_REPORTEVENT
- PREEMPTIVE_OS_REVERTTOSELF
- PREEMPTIVE_OS_RSFXDEVICEOPS
- PREEMPTIVE_OS_SECURITYOPS
- PREEMPTIVE_OS_SERVICEOPS
- PREEMPTIVE_OS_SETENDOFFILE
- PREEMPTIVE_OS_SETFILEPOINTER
- PREEMPTIVE_OS_SETFILEVALIDDATA
- PREEMPTIVE_OS_SETNAMEDSECURITYINFO
- PREEMPTIVE_OS_SQLCROPS
- PREEMPTIVE_OS_SQMLAUNCH
- PREEMPTIVE_OS_VERIFYSIGNATURE
- PREEMPTIVE_OS_VERIFYTRUST
- PREEMPTIVE_OS_VSSOPS
- PREEMPTIVE_OS_WAITFORSINGLEOBJECT
- PREEMPTIVE_OS_WINSOCKOPS
- PREEMPTIVE_OS_WRITEFILE
- PREEMPTIVE_OS_WRITEFILEGATHER
- PREEMPTIVE_OS_WSASETLATEROR
- PREEMPTIVE_REENLIST
- PREEMPTIVE_RESIZELOG
- PREEMPTIVE_ROLLFORWARDREDO
- PREEMPTIVE_ROLLFORWARDUNDO
- PREEMPTIVE_SB_STOPENDPOINT
- PREEMPTIVE_SERVER_STARTUP
- PREEMPTIVE_SETRMINFO
- PREEMPTIVE_SHAREDMEM_GETDATA
- PREEMPTIVE_SNIOPEN
- PREEMPTIVE_SOSHOST
- PREEMPTIVE_SOSTESTING
- PREEMPTIVE_STARTRM
- PREEMPTIVE_STREAMFCB_CHECKPOINT
- PREEMPTIVE_STREAMFCB_RECOVER
- PREEMPTIVE_STREEDRIVER
- PREEMPTIVE_TESTING
- PREEMPTIVE_TRANSIMPORT
- PREEMPTIVE_UNMARSHAKPROPAGATIONTOKEN
- PREEMPTIVE_VSS_CREATESNAPSHOT
- PREEMPTIVE_VSS_CREATEVOLUMESNAPSHOT
- PREEMPTIVE_XE_CALLBACKEXECUTE
- PREEMPTIVE_XE_DISPATCHER
- PREEMPTIVE_XE_DENGINEINIT
- PREEMPTIVE_XE_GETTARGETSTATE

- PREEMPTIVE_XESESSIONCOMMIT
- PREEMPTIVE_TARGETFINALIZE
- PREEMPTIVE_XE_TARGETINIT
- PREEMPTIVE_XE_TIMERRUN
- PREEMPTIVE_XETESTING
- PRINT_ROLLBACK_PROGRESS
- PWAIT_ALL_COMPONENTS_INITIALIZED
- QDS_ASYNC_PERSIST_TASK_START
- QDS_PERSIST_TASK_MAIN_LOOP_SLEEP
- QDS_SHUTDOWN_QURUR
- QDS_TASK_START
- QNMANAGER_ACQUIRE
- QPJOB_KILL
- QPJOB_WAITFOR_ABORT
- QRY_MEM_GRANT_INFO_MUTEX
- QUERY_ERRHDL_SERVICE_DONE
- QUERY_EXECUTION_INDEX_SORT_EVENT_OPEN
- QUERY_NOTIFICATION_MGR_MUTEX
- QUERY_NOTIFICATION_SUBSCRIPTION_MUTEX
- QUERY_NOTIFICATION_TABLE_MGR_MUTEX
- QUERY_NOTIFICATION_UNITTEST_MUTEX
- QUERY_OPTIMIZER_PRINT_MUTEX
- QUERY_TRACEOUT
- QUERY_WAIT_ERRHDL_SERVICE
- RECOVER_CHANGEDB
- REPL_CACHE_ACCESS
- REPL_HISTORYCACHE_ACCESS
- REPL_SCHEMA_ACCESS
- REPL_TRANHASHTABLE_ACCESS
- REPLICA_WRITES
- REQUEST_DISPENSER_PAUSE
- RESERVED_MEMORY_ALLOCATION_EXT
- RESMGR_THROTTLED
- RFS_MSGPUMP
- RFS_NOTIFYTHREAD
- RFS_STARTUP
- RG_RECONFIG
- SEC_DROP_TEMP_KEY
- SECURITY_CRYPTO_CONTEXT_MUTEX
- SECURITY_MUTEX
- SEQUENTIAL_GUID
- SERVER_IDLE_CHECK
- SHUTDOWN
- SLEEP_BPOOL_FLUSH
- SLEEP_DBSTARTUP
- SLEEP_DCOMSTARTUP
- SLEEP_MASTERDBREADY
- SLEEP_MSDBSTARTUP
- SLEEP_TEMPDBSTARTUP
- SNI_CRITICAL_SECTION
- SNI_HTTP_ACCEPT
- SNI_HTTP_WAITFOR_0_DISCON
- SNI_LISTENER_ACCESS

- SNI_TASK_COMPLETION
- SOAP_READ
- SOAP_WRITE
- SOS_CALLBACK_REMOVAL
- SOS_DISPATCHER_MUTEX
- SOS_LOCALALLOCATORLIST
- SOS_MEMORY_USAGE_ADJUSTMENT
- SOS_OBJECT_STORE_DESTROY_MUTEX
- SOS_PHYS_PAGE_CACHE
- SOS_PROCESS_AFFINITY_MUTEX
- SOS_SCHEDULER_YIELD
- SOS_SMALL_PAGE_ALLOC
- [SOS_STACKSTORE_INIT_MUTEX](#)
- SOS_SYNC_TASK_ENQUEUE_EVENT
- SOS_VIRTUALMEMORY_LOW
- SOSHOST_EVENT
- SOSHOST_INTERNAL
- SOSHOST_MUTEX
- SOSHOST_RWLOCK
- SOSHOST_SEMAPHORE
- SOSHOST_SLEEP
- SOSHOST_TRACELOCK
- SOSHOST_WAITFORDONE
- SQLCLR_APPDOMAIN
- SQLCLR_ASSEMBLY
- SQLCLR_DEADLOCK_DETECTION
- SQLCLR_QUANTUM_PUNISHMENT
- SQLSORT_NORMMUTEX
- SQLSORT_SORTMUTEX
- SQLTRACE_FILE_BUFFER
- SQLTRACE_FILE_READ_IO_COMPLETION
- SQLTRACE_FILE_WRITE_IO_COMPLETION
- SQLTRACE_INCREMENTAL_FLUSH_SLEEP
- SQLTRACE_LOCK
- SQLTRACE_SHUTDOWN
- SQLTRACE_WAIT_ENTRIES
- SRVPROC_SHUTDOWN
- TEMPOBJ
- [THREADPOOL](#)
- TIMEPRIV_TIMEPERIOD
- TRACE_EVTNOTIF
- TRACEWRITE
- TRANSACTION_MUTEX
- UTIL_PAGE_ALLOC
- VIA_ACCEPT
- VIEW_DEFINITION_MUTEX
- WAIT_FOR_RESULTS
- WAIT_XTP_HOST_WAIT
- WAITFOR_TASKSHUTDOWN
- WAITSTAT_MUTEX
- WCC
- WORKTBL_DROP
- WRITE_COMPLETION

- XACT_OWN_TRANSACTION
- XACT_RECLAIM_SESSION
- XACTWORKSPACE_MUTEX
- XE_BUFFERMGR_ALLPROCESSED_EVENT
- XE_BUFFERMGR_FREEBUF_EVENT
- XE_DISPATCHER_CONFIG_SESSION_LIST
- XE_DISPATCHER_JOIN
- XE_MODULEMGR_SYNC
- XE_OLS_LOCK
- XE_PACKAGE_LOCK_BACKOFF
- XE_SERVICES_EVENTMANUAL
- XE_SERVICES_MUTEX
- XE_SERVICES_RWLOCK
- XE_SESSION_CREATE_SYNC
- XE_SESSION_FLUSH
- XE_SESSION_SYNC
- XE_STM_CREATE
- XE_TIMER_MUTEX
- XE_TIMER_TASK_DONE

CXPACKET wait type actions

SQL Server waits provide information about what the database engine is waiting on while extracting queries on a monitored server. While there are many different wait types, CXPACKET waits are a commonly misinterpreted data measurement and require additional clarification.

CXPACKET waits often result when parallelized query threads have an unequal workload, causing a block on the entire query until the longest-running thread is complete. Note that the trigger for the CXPACKET wait occurs not with parallelism, but with the distribution of work on the individual threads.

While you can decrease the number of CXPACKET waits by reducing the MAXDOP on the monitored server, this is often not a recommended course of action as it may cause the wait to disappear and the underlying query to take longer to execute. For example, this issue may result from large parallel table scans, which you can verify using other wait types on the server, such as PAGIOLATCH_XX. This may also result from out-of-date statistics that cause a bad query plan and the unequal workload among threads.

Before adjusting MAXDOP, review your indexes and statistics for accuracy and make sure there are no other underlying issues. If you choose to adjust MAXDOP, recommended settings include 4, 2, or 1. For a NUMA system, use the number of cores available in a single NUMA node. You may want to make these updates on the single query instead of the entire instance to avoid any additional problems.

LOGMGR_RESERVE_APPEND

Occurs when a task is waiting to see whether log truncation frees up log space to enable the task to write a new log record. Consider increasing the size of the log file(s) for the affected database to reduce this wait.

SOS_STACKSTORE_INIT_MUTEX

This wait type occurs during the synchronization of internal store initialization. Look for systems of high blocking or contention problems with many of the workers and if detected, consider increasing the max working threads if you do not use the default setting of 0.

THREADPOOL

Look for symptoms of high blocking or contention problems with workers if the wait count and times are high. Consider increasing the max worker thread only if the default setting is anything but zero.

Occurs when a task is waiting to be run on. This can indicate that the maximum worker setting is too low, or that batch executions are taking unusually long, thus reducing the number of workers available to satisfy other batches.

Wait type category - Transaction Log

The Transaction Log wait type category includes all the waits that are associated with transaction log bottlenecks, including the following:

- [LOGBUFFER](#)
- LOGMGR
- WRITELOG

LOGBUFFER

This wait type is typically encountered whenever there is an I/O bottleneck. Look for WRITERLOG waiters and if found the overall problem is an I/O bottleneck on the storage system associated with the transaction log.

Metric alerts

SQL Diagnostic Manager offers numerous alerts to successfully monitor and warn you about your SQL Server instances and availability groups. Proper configuration is key to making sure you receive alerts only on those metrics that actually fall outside the typical performance of your environment. SQL Diagnostic Manager includes the following metric alerts for your use:

 The alerts are listed by default Ranks and categories. The default Ranks are classified as follows:

- Rank 10 = Service High
- Rank 20 = Resource High
- Rank 30 = Database High
- Rank 40 = Process High
- Rank 50 = Service Low
- Rank 60 = Resource Low
- Rank 70 = Database Low
- Rank 80 = Process Low
- Rank 90 = Other

Category: Backup

Metric	Rank
Days since last backup	40

Category: Databases

Metric	Rank
Availability Group Estimated Data Loss (Seconds)	20
Availability Group Estimated Recovery time (Seconds)	20
Availability Group Log Send Queue Size (KB)	20
Availability Group Preferred Replica	10
Availability Group Redo Queue Size (KB)	20
Availability Group Redo Rate (KB/sec)	20
Availability Group Role Change	10
Availability Group Synchronization Health	20

Metric	Rank
Availability Group Synchronization Performance (Seconds)	20
Data File Autogrow	70
Database Full (Percent)	30
Database Full (Size)	30
Database Read/Write Error Occurred	30
Database Status	30
Filegroup Space Full (Percent)	30
Filegroup Space Full (Size)	30
Index Row Hits (Percent)	70
Log File Autogrow	70
Log Full (Percent)	30
Log Full (Size)	30
Longest Running Version Store Transaction (Minutes)	30
Mirror Commit Overhead (Milliseconds)	30
Mirrored Server Role Change	30
Mirroring Oldest Unsent Transactions	30
Mirroring Preferred Configuration	30
Mirroring Status	30
Mirroring Unrestored Log (KB)	30
Mirroring Unsent Log (KB)	30

Metric	Rank
Mirroring Witness Connection	30
Session Tempdb Space Usage (MB)	80
Table Fragmentation (Percent)	70
Tempdb Contention (ms)	20
Version Store Generation Ratio	30
Version Store Size (MB)	30

Category: DBaaS

Metric	Rank
Average Data IO Percent High	100
Average Data IO Percent Low	100
Average Log Write Percent High	100
Average Log Write Percent Low	100
CPU Credit balance	100
CPU Credit Usage	100
Database Average Memory Usage Percent High	100
Database Average Memory Usage Percent Low	100
Disk Queue Depth	100
In-Memory Storage Usage Percent High	100
In-Memory Storage Usage Percent Low	100
Max Session Percent High	100

Metric	Rank
Max Session Percent Low	100
Max Worker Percent High	100
Max Worker Percent Low	100
Read Latency	100
Read Throughput	100
Service Tier Changes	100
Swap Usage	100
Write Latency	100
Write Throughput	100

Category: Logs

Metric	Rank
SQL Server Agent Log	40
SQL Server Error Log	40

Category: Operational

Metric	Rank
CLR Enabled	90
OLE Automation Disabled	90
OS Metrics Collection Status	90
SQL Server Agent XPs Disabled	50

Metric	Rank
SQL Server Data Used (Percent)	30
SQL Server Log Used (Percent)	30
SQLdm Repository Grooming Timed Out	90
WMI Service Unavailable	50

Category: Queries

Metric	Rank
Query Monitor Events (Counts)	80

Category: Resources

Metric	Rank
Average Disk Milliseconds Per Read	20
Average Disk Milliseconds Per Transfer	20
Average Disk Milliseconds Per Write	20
Disk Reads Per Second	60
Disk Transfers Per Second	60
Disk Writes Per Second	60
OS Average Disk Queue Length (Count)	20
OS Average Disk Queue Length Per Disk (Count)	60
OS Disk Free Space (Size)	20
OS Disk Full (Percent)	20

Metric	Rank
OS Disk Time (Percent)	60
OS Disk Time Per Disk (Percent)	60
OS Memory Usage (Percent)	20
OS Paging (Per Second)	60
OS Privileged Time (Percent)	60
OS Processor Queue Length (Count)	20
OS Processor Time (Percent)	20
OS User Time (Percent)	60
Page Life Expectancy	60
Procedure Cache Hit Ratio	20
SQL Server CPU Usage (Percent)	20
SQL Server Memory Usage (Percent)	20

Category: Services

Metric	Rank
Cluster Active Node	10
Cluster Failover	10
Distribution Latency (Seconds)	40
DTC Status	10
Full-Text Search Status	50

Metric	Rank
Last Full-Text Catalog Update (Hours)	50
Latent replication transaction (Count)	40
SQL Server Active Directory Helper Service Status	50
SQL Server Agent Job Completion	80
SQL Server Agent Job Failure	40
SQL Server Agent Long Running Job (Minutes)	80
SQL Server Agent Long Running Job (Percent)	80
SQL Server Agent Status	10
SQL Server Browser Service Status	50
SQL Server Status	10
Unsubscribed Transactions (Count)	40
Unsubscribed Transactions (Seconds)	80

Category: Sessions

Metric	Rank
Blocked Sessions	40
Blocked Sessions Wait Time (Seconds)	40
Client Computers	60
Deadlock	80
Oldest Open Transaction (Minutes)	40

Metric	Rank
Session CPU Time (Seconds)	80
SQL Server Response Time (Milliseconds)	40
User Connections (Percent)	20

Category: Virtualization

Metric	Rank
Host CPU Usage (Percent)	20
Host Memory Swap Detected	20
Host Memory Usage (Percent)	20
Host Power State	20
VM CPU Ready Wait Time (ms)	20
VM CPU Usage (Percent)	20
VM Host Server Change	20
VM Memory Swap Delay Detected	20
VM Memory Usage (Percent)	20
VM Power State	20
VM Reclaimed/Ballooned Memory (KB)	20
VM Resource Configuration Change	20
VM Resource Limits Detected	20

Availability Group Estimated Data Loss (Seconds) alert

The Availability Group Estimated Data Loss (Seconds) alert tracks the time period in seconds that data is not synchronized to the secondary replica. This value reflects the time difference of the last commit between the primary and secondary replicas. Alert includes affected databases.

For more information on configuring alerts and setting thresholds, see [Configure alerts](#) .



Enable baseline thresholds

To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.



Create an alert response bundle

Create an alert response bundle with the Availability Group Estimated Data Loss (Seconds) alert and related alerts. For additional information, see [Create alert response bundles](#).

Availability Group Estimated Recovery time (Seconds) alert

The Availability Group Estimated Recovery time (Seconds) alert provides the required time in seconds that the secondary replica needs to catch up with the primary replica. Alert includes affected databases.

For more information on configuring alerts and setting thresholds, see [Configure alerts](#) .

✔ Enable baseline thresholds

To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

✔ Create an alert response bundle

Create an alert response bundle with the Availability Group Estimated Recovery time (Seconds) alert and related alerts. For additional information, see [Create alert response bundles](#).

Availability Group Log Send Queue Size (KB) alert

The Availability Group Log Send Queue Size (KB) alert indicates the amount of log records in KB needed to ship to the secondary replica to complete synchronization. Alert includes affected databases.

For more information on configuring alerts and setting thresholds, see [Configure alerts](#).



Enable baseline thresholds

To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.



Create an alert response bundle

Create an alert response bundle with the Availability Group Log Send Queue Size (KB) alert and related alerts. For additional information, see [Create alert response bundles](#).

Availability Group Preferred Replica alert

The Availability Group Preferred Replica alert triggers when the preferred replica defined for a database changes to a different replica in a set period of time.

For more information on configuring alerts and setting thresholds, see [Configure alerts](#).

- ✔ Create an alert response bundle with the Availability Group Preferred Replica alert and related alerts. For additional information, see [Create alert response bundles](#).

Availability Group Redo Queue Size (KB) alert

The Availability Group Redo Queue Size (KB) alert indicates the amount of log records from log files in KB that need redoing in the secondary replica to complete synchronization. Alert includes affected databases.

For more information on configuring alerts and setting thresholds, see [Configure alerts](#) .



Enable baseline thresholds

To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.



Create an alert response bundle

Create an alert response bundle with the Availability Group Redo Queue Size (KB) alert and related alerts. For additional information, see [Create alert response bundles](#).

Availability Group Redo Rate (KB\sec) alert

The Availability Group Redo Rate (KB\sec) alert provides the rate in KB per second at which log records are redone in the secondary database to complete synchronization. Alert includes affected databases.

For more information on configuring alerts and setting thresholds, see [Configure alerts](#).

✔ Enable baseline thresholds

To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

✔ Create an alert response bundle

Create an alert response bundle with the Availability Group Redo Rate (KB\sec) alert and related alerts. For additional information, see [Create alert response bundles](#).

Availability Group Role Change alert

The Availability Group Role Change alert indicates when the primary role changes to a secondary replica. Alert includes affected databases for the entire availability group.

For more information on configuring alerts and setting thresholds, see [Configure alerts](#).

- ✔ Create an alert response bundle with the Availability Group Role Change alert and related alerts. For additional information, see [Create alert response bundles](#).

Availability Group Synchronization Health alert

The Availability Group Synchronization Health alert provides the database synchronization state in the availability group. Possible values include: not healthy, partially healthy, and healthy.

For more information on configuring alerts and setting thresholds, see [Configure alerts](#).

- ✔ Create an alert response bundle with the Availability Group Synchronization Health alert and related alerts. For additional information, see [Create alert response bundles](#).

Availability Group Synchronization Performance (Seconds) alert

The Availability Group Synchronization Performance (Seconds) alert indicates the time required in seconds to complete synchronization between the primary and secondary replicas.

For more information on configuring alerts and setting thresholds, see [Configure alerts](#).

✔ Enable baseline thresholds

To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

✔ Create an alert response bundle

Create an alert response bundle with the Availability Group Synchronization Performance (Seconds) alert and related alerts. For additional information, see [Create alert response bundles](#).

Average Data IO Percent High

This metric is the high percentage of Data IO being used to the current service tier of the database.

Average Data IO Percent Low

The average Data IO Percent Low metric is the percentage of Data IO being used relative to the current service tier of the database. Consistent low values indicate that the service tier selected may be more than the average Log Write Percent High what is required for the database.

Average Disk Milliseconds Per Read alert

The Average Disk Milliseconds Per Read alert provides the average time (in milliseconds) that a read operation uses data from the disk.

Average Disk Milliseconds Per Transfer alert

The Average Disk Milliseconds Per Transfer alert provides the average time (in milliseconds) used by a disk transfer operation.

Average Disk Milliseconds Per Write alert

The Average Disk Milliseconds Per Write alert provides the average time (in milliseconds) used by an operation writing data to the disk.

Average Log Write Percent High

The Average Log Write Percent High metric is the percentage of log write activity being used relative to the current service tier of the database.

Average Log Write Percent Low

The Average Log Write Percent Low metric alert is the percentage of log write activity being used relative to the current service tier of the database. Consistent low values indicate that the service tier selected may be more than what is required for the database.

Blocked Sessions (Count) alert

The Blocked Sessions (Count) alert provides the number of sessions on the SQL Server instance blocked by other sessions holding requested locks. For this alert to display, enable monitoring of non-query activities with capturing of blocking events.

By default, blocked sessions wait for a lock wait forever, which is not optimum behavior. You can change this default behavior by making a connection, and then adding `SET LOCKTIMEOUT 60000` to the top of the first T-SQL batch.

Reduce the number of blocked sessions

If you experience more than one blocked session on an average day, your site may experience one of the following issues:

- T-SQL batches submitted containing a `BEGIN TRANSACTION` statement with no corresponding `COMMIT TRANSACTION` statement. You must correct the T-SQL.
- T-SQL batches submitted containing a `BEGIN TRANSACTION` statement but where the `COMMIT TRANSACTION` statement is in a following T-SQL batch that is only executed once the end-user confirms the transaction. You must correct the T-SQL.
- The site includes some long-running transactions processing at peak times.

When your transactions run longer than they should and clash, consider drastically reducing the time that a lock is held by:

- Doing as much work as possible (for example, `SELECT` statements that may be needed) before the transaction performs its first update, delete, or insert.
- Group all `UPDATES`, `DELETES`, and `INSERTS` as closely as possible together within a transaction with as few selects as possible separating them.
- Commit the transaction as soon as possible after the last DML statement.
- Once the transaction has begun do not have any stops for user input. Gather all user inputs before the transaction starts.
- Avoid the use of server side cursors during a transaction as they slow execution considerably.
- If a stored procedure and/or trigger are invoked inside a transaction minimize or eliminate the number of SQL re-compilations made to that object. See the SQL Re-compilations counter for ways to dramatically reduce recompiles.
- Increase the speed of transaction throughput such that it becomes less likely that one transaction waits for the preceding one by:
 - Adding more disks to your RAID solution.
 - Replacing your disks with faster disks.
 - Switching your RAID array from a RAID 5 to a RAID 10 solution as each write IO results in two writes using RAID 10 vs. four for RAID 5 (100% more efficient with writes).
 - So long as the RAID controller has some form of battery backup, then switch its cache mode from Write-through to Write-back as this increases the system's ability to handle write IOs by an order of magnitude.
 - Adding more cache memory to the RAID controller.
 - Adding more RAM to the server.
 - Adding another CPU to a SMP computer.
 - Upgrading the CPU, memory and motherboard with faster models.
 - Minimize the number of Context Switches by turning on Use NT Fibers in SQL Server.
 - Switch the Boost SQL Server Priority on.

 **Enable baseline thresholds**

To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

 **Create an alert response bundle**

Create an alert response bundle with the Blocked Sessions (Count) alert and related alerts. For additional information, see [Create alert response bundles](#).

Blocking Session Wait Time (Seconds) alert

The Blocking Session Wait Time (Seconds) alert provides the time in seconds that a SQL Server session blocks other sessions.

By default, locked sessions waiting for a lock wait forever, which is not optimum behavior from an end-user point of view. You can change this default behavior by adding `SET LOCKTIMEOUT 60000` to the top of the first T-SQL batch after making a connection.

The [Active Alerts view](#) provides additional information on the Blocking Session Wait Time (Seconds) alert. To access this information, refer to the right-click context menu after having organized your alert view by metric. This menu grants access to real time alert information, historical view, block details, and online help specific to the alert.

- ✔ To access similar information on the [Blocking view](#), select a specific SQL Server instance, point to Sessions, and then select the Blocking ribbon.

Blocking Sessions Report

The Blocking Sessions Report displays blocking and blocked sessions' information. To access the Blocking Sessions Report in the Alert view, select the **Show Block Details** option in the right-click context menu.

Key information provided for blocking and blocked sessions includes:

Blocking Process Details:

- Session ID
- Hostname
- User Name
- Application
- Database
- Last Batch Started
- Transaction ID
- Open Transactions

Blocked Process Details:

- Session ID
- Hostname
- User Name
- Application
- Database
- Last Batch Started
- Transaction ID
- Open Transactions
- Wait Time (ms)
- Wait Resource

- ✔ To retrieve the blocking sessions report for SQL Diagnostic Manager, select the **"Non-Query activities>Capture Blocking (SQL 2005+)"** check box in the Activity Monitor Tab.

Reduce blocked sessions wait time

If you experience more than one blocked session on an average day, your site may experience one of the following issues:

- T-SQL batches submitted containing a `BEGIN TRANSACTION` statement with no corresponding `COMMIT TRANSACTION` statement. You must correct the T-SQL.
- T-SQL batches submitted containing a `BEGIN TRANSACTION` statement but where the `COMMIT TRANSACTION` statement is in a following T-SQL batch that is only executed once the end-user confirms the transaction. You must correct the T-SQL.
- The site includes some long-running transactions processing at peak times.

Where your transactions run longer than they should and clash, consider drastically reducing the time that a lock is held by:

- Performing as much work as possible before the transaction performs its first update, delete, or insert. For example, add any necessary `SELECT` statements.
- Grouping all `UPDATES`, `DELETES`, and `INSERTS` as closely as possible within a transaction with as few `SELECTS` as possible separating them.
- Committing the transaction as soon as possible after the final DML statement.
- Avoiding any stops for user input once the transaction begins. Be sure to gather all user inputs before the transaction starts.
- Avoiding the use of server-side cursors during a transaction as they slow execution considerably.
- Minimizing or eliminating the number of SQL re-compilations made to the object if a stored procedure and/or trigger is invoked inside a transaction. See the SQL RE-compilations counter for steps to dramatically reduce recompiles.
- Increasing the speed of transaction throughput such that it becomes less likely that one transaction waits for the preceding transaction. You can improve transaction throughput speed by:
 - Adding more disks to your RAID solution.
 - Replacing your disks with faster disks.
 - Switching your RAID array from a RAID 5 to a RAID 10 solution. Note that each write IO results in two writes using RAID 10 vs. four for RAID 5 (100% more efficient with writes).
 - Switching the RAID controller cache mode from Write-through to Write-back so long as the RAID controller has some form of battery backup. This change increases the system's ability to handle write IOs by an order of magnitude.
 - Adding more cache memory to the RAID controller.
 - Adding more RAM to the server.
 - Adding another CPU to the SMP computer.
 - Upgrading the CPU, memory, and motherboard with faster models.
 - Minimizing the number of Context Switches by turning on Use NT Fibers in SQL Server.
 - Switching on the Boost SQL Server Priority.



Create an alert response bundle

Create an alert response bundle with the Blocking Session Wait Time (Seconds) alert and related alerts. For additional information, see [Create alert response bundles](#).

Client Computers (Count) alert

The Client Computers (Count) alert provides the number of unique client computers connected to the SQL Server instance. SQL Diagnostic Manager measures this value by getting the count of processes from distinct computers in sysprocesses table.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

CLR Enabled alert

The CLR Enabled alert indicates that the Common Language Runtime (CLR) configuration option is enabled for the SQL Server instance. You can run managed code in the .NET Framework. Use the Server Configuration view to reconfigure this option on SQL Server 2005 instances.

Cluster Active Node alert

The Cluster Active Node alert indicates that the active node of the cluster is non-preferred. The active cluster node changes when a failover event occurs. If the original preferred node is offline for an extended period of time, you can change the preferred node setting by right-clicking the instance in the **Servers** tree and selecting **Properties > Cluster Settings**. By default, the preferred node is first node encountered when monitoring began.

Cluster Failover alert

The Cluster Failover alert indicates a change in the active node. This alert remains active for the amount of time specified in the alert's advanced settings.

To edit these settings, right-click the server in the Servers tree and select Configure Alerts. Click the **Advanced** button on the Cluster Failover Alert and edit the time.

Prevent cluster failover alerts when you have a scheduled failover

You can temporarily put the server into maintenance mode by right-clicking the server in the Servers tree and selecting Maintenance mode. When the scheduled failover is over, right-click the instance in the **Servers** tree and select **Disable Maintenance Mode**.

CPU Credit Balance

The CPU Credit Balance metric is relevant to RDS databases in a T2 level instance. The more vCPUs available to the instance, the higher the number of credits per hour should be.

CPU Credit Usage

The CPU Credit Usage is relevant to RDS databases running in a T2 level instance. The more vCPUs available to the instance, more credits per hour should accumulate.

Database Average Memory Usage Percent High

The Database Average Memory Usage Percent High metric is the percentage of memory used for the database, relative to the maximum available for the service tier.

Database Average Memory Usage Percent Low

The Database Average Memory Usage Percent metric is the percentage of memory used for the database, relative to the maximum available for the service tier. Consistently low values indicate that the service tier may be more than what is required for the workload.

Database Full (Percent) alert

The Database Full (Percent) alert provides the percentage of the allowable disk space for the database currently used by the sum of data, text, and indexes. Allowable disk space for the database is calculated by taking into account the current allocated space, SQL Diagnostic Manager auto-growth settings and available disk space. If this value exceeds 80%, consider replacing disks with bigger disks.

This alert is raised based on the database metrics collection interval specified in Server Properties. For more information, see [Set general server options](#).

Calculate the database full percentage using the autogrow settings

When setting the thresholds for your Database Full (Percent) alert, click **Advanced > Autogrow Settings** and SQL Diagnostic Manager provides a choice for you to calculate your displayed percentage. Select **Yes, alert on the current used size divided by the maximum possible size** to take possible autogrowth into consideration when calculating the database full percentage. Select **No, alert on the current used size divided by the current file size** to calculate the database full percentage based only on the current size of the database disk file.

- ✔ Create an alert response bundle with the Database Full (Percent) alert and related alerts. For additional information, see [Create alert response bundles](#).

Database Full (Size) alert

The Database Full (Size) alert provides the amount of disk space, in megabytes, currently used by the database as the sum of data, text, and indexes. This value represents the actual amount of used data within the database file and does not represent the size of the database file itself.

SQL Diagnostic Manager raises this alert based on the database metrics collection interval specified in Server Properties. For more information, see [Set general server options](#).

- ✔ Create an alert response bundle with the Database Full (Size) alert and related alerts. For additional information, see [Create alert response bundles](#).

Database Read/Write Error Occurred alert

The Database Read/Write Error Occurred alert indicates that an I/O error occurred while attempting to read from or write to a database file. If this error persists consider the following solutions:

- Replace the disks.
- File may be corrupted, see if you can restore from a backup.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

Database Status alert

The Database Status alert indicates that the [Database Status metric](#) reached an alert status. This status results when the database:

- Is inaccessible
- Has an issue with loading, crashed while loading, has not recovered, in recovery, suspect, or in emergency mode
- Contains databases sizes in the error threshold
- Contains logs with sizes in the error threshold
- Contains tables with reorganization needs in the error threshold
- Is transitioning, loading, being check pointed, set as DBO use only, or in single-user mode

This alert is raised based on the diagnostic data and database metrics collection intervals specified in Server Properties. For more information, see [Set general server options](#).

Reduce the number of database status issues

Navigate to the **Summary** view on the **Database** tab of the SQLDM Console. Look for databases that have a critical or warning status and review what the issue is (the problem is highlighted either in yellow or red). You may need to rebuild an index, increase allocated file size, or turn on the auto-growth feature for your database or logs.

Data File Autogrow alert

The Data File Autogrow alert indicates that a data file autogrow occurred on the specified database. For this alert to display, enable monitoring of non-query activities with capturing of autogrow events. Note that you can avoid unnecessarily long reads and poor response time from SQL Server by minimizing the number of autogrow events occurring in your environment.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

To learn more about how autogrow works and related best-practices, see [Microsoft Support Article 315512](#), "Considerations for the 'autogrow' and 'autoshrink' settings in SQL Server."

- ✔ Create an alert response bundle with the Data File Autogrow alert and related alerts. For additional information, see [Create alert response bundles](#).


Days since last backup alerts

The Days since last backup alerts indicate the number of days that databases ([per-database](#) or [at the instance level](#)) have not been backed up.

To see more information about these alerts applied to Availability Groups, check [Days since last backup for availability groups](#)

Days since last backup at the instance-level


The Days since last backup alert applies to any database at the instance level.

 The Days since last backup alert is enabled by default.

To configure thresholds and employ alert filters such as excluding specific databases, follow these steps:

To configure an instance-level alert:

1. Right-click a SQL Server instance in the Servers tree.
2. Select **Configure Alerts**.
3. Select the Days since last backup metric in the **Alert Configuration** window.
4. Click **Edit to** modify default threshold (number of days).
5. Check the boxes next to **Informational**, **Warning**, and **Critical** to include alerts for these states.
6. Change the alert thresholds by moving the arrows to the appropriate levels or by double-clicking the value and typing a new threshold level.
7. Click **Advanced** to apply advanced settings such as [alert suppression](#) or alert filters ([excluding a database from alerts](#) or/and [Availability Group Collection](#)).
8. Click **OK** to accept your changes.

 The configuration of the "Days since last backup" alert now has the option to include or exclude Read-Only databases.

Days since last backup per-database

The Days since last backup alert applies to a specific database in your monitored SQL Server instance. This alert is disabled by default.

To configure thresholds and employ alert filters such as excluding specific databases, follow these steps:

To configure a per-database alert:

1. Right-click a SQL Server instance in the Servers tree.
2. Select **Configure Alerts**.
3. Select the Days since last backup metric in the **Alert Configuration** window.
4. **If the metric features a per-database**, on the Configuration tab of the Alert Configuration window, click **Add**. Use the drop-down list to select the database to which you want to apply these settings.
5. Check the boxes next to **Informational**, **Warning**, and **Critical** to include alerts for these states.
6. Change the alert thresholds by moving the arrows to the appropriate levels or by double-clicking the value and typing a new threshold level.
7. Click **Advanced** to apply [alert suppression](#).
8. Click **OK** to accept your changes.

Days since last backup for availability groups

The Advanced Alert Configuration for this metric includes the following options for Availability Group Collection Options:

- Primary
- Secondary
- Default
- Both

And under the Replica Collections Options

- Monitored Only
- All

Here are some scenarios applied to Availability Groups:

- **When primary and (any or all) other secondary servers are being monitored:** Primary server displays the last backup date alert by checking all other secondary servers for the latest backup date of that database. If the latest backup date across that Availability Groups justifies the alert, it displays. The secondary monitored databases also display the same alert, as the latest backup date for an Availability Group across all replicas will be the same.
- **When only primary replica of a database is being monitored:** In this scenario the alert displays after SQLDM checks the latest backup date across all replicas and if that date justifies the displayed alert.
- **When some secondary replica is being monitored but no primary replica:** In this scenario the alert displays after SQLDM checks the latest backup date across all replicas and if that date justifies the displayed alert.

Limitations

- The assumption is that the connection credentials of all the replicas are the same. At this point SQLDM stores connection credentials only of the monitored servers. If a secondary replica is not monitored, the connection credentials are unknown. This setting can not be included in the properties as a number of secondary replicas because is not limited and also its db specific.
- A replica may have the latest backup date, but it is possible that the sync is broken between replicas, in which case the db would have been backed up but the some data would not be there. In this case, the alert may not come if the last back update does not justify an alert.

Deadlock alert

The Deadlock alert indicates that a deadlock occurred on the monitored server. For this alert to display, enable monitoring of non-query activities with capturing of deadlock events.

Deadlocks occur when a process or several processes attempt to access a resource that the other holds a lock on. Neither task can complete because both processes are attempting to lock the same resource.

SQL Server chooses one or more deadlock to terminate so that processing can continue.

Resolve deadlocks

When a Deadlock Alert occurs, check the Alerts view in SQL Diagnostic Manager and select a deadlock alert. The SPID, Hostname, Login Name, Client Application, and Database are listed.

For more information, select **Show Deadlock Details**. This information helps with diagnosing the problem application and includes the option to export deadlock data to view in SQL Server Management Studio or SQL Server Profiler.

- ✔ Create an alert response bundle with the Deadlock alert and related alerts. For additional information, see [Create alert response bundles](#).

Disk Queue Depth

Disk Queue Depth is a measure of how many outstanding requests are waiting to be handled by the storage system.

Disk Reads Per Second alert

The Disk Reads Per Second alert provides the rate of read operations on the disk. A high number of disk reads may indicate that SQL server is executing poorly performing queries. Note that this alert is disk specific.

This alert is raised based on the diagnostic data collection interval specified in [Server Properties](#). Note that you need to enable this alert first in the Alert Configuration window to rise.

✓ The [Disk throughput chart](#) provides a graphical analysis of the number of disk reads and writes over time.

Disk Transfers Per Second alert

Comprised of both Disk Reads/sec and Disk Writes/sec, the Disk Transfers Per Second alert provides the rate of transfer operations on the disk and helps to determine if a drive does not have enough supporting disks. Note that this alert is disk specific.

A large increase in disk transfers per second may signal a change in workload or a developing I/O or memory pressure issue.

This alert is raised based on the diagnostic data collection interval specified in [Server Properties](#). Note that you need to enable this alert first in the Alert Configuration window to rise.

✔ For additional information on the Disk Transfers/sec counter, see [Monitoring Disk Usage](#).

Disk Writes Per Second alert

The Disk Writes Per Second alert provides the rate of write operations on the disk. Note that this alert is disk specific.

A high number of disk writes per second may indicate that the SQL Server is running low on allocated memory due to large amounts of data being written. It may also point to a situation of memory pressure that occurs when data is not able to remain in cache and has to be flushed to disk frequently.

This alert is raised based on the diagnostic data collection interval specified in [Server Properties](#). Note that you need to enable this alert first in the Alert Configuration window to rise.

✔ The [Disk throughput chart](#) provides a graphical analysis of the number of disk reads and writes over time.

Distribution Latency (Seconds) alert

The Distribution Latency (Seconds) alert tracks the amount of time, in seconds, that a replication transaction is at the Publisher waiting for a Distributor to receive it, and has exceeded the warning threshold.

Causes of high distribution latency

The most common cause of high distribution latency is when a transaction has a large amount of commands. Once the transaction is complete, distribution latency decreases.

The **Replication** view on the **Services** tab displays a graph with the non-distributed count and the non-distributed time. Graphs that trend upward may indicate a problem that needs resolution.

Reduce distribution latency

You can reduce distribution latency by:

- Reducing traffic on the publisher
- Limiting non-replication related work by the publisher
- Increasing the network bandwidth available to the publisher

DTC Status alert

The DTC Status alert provides the status of the Distributed Transaction Coordinator service.

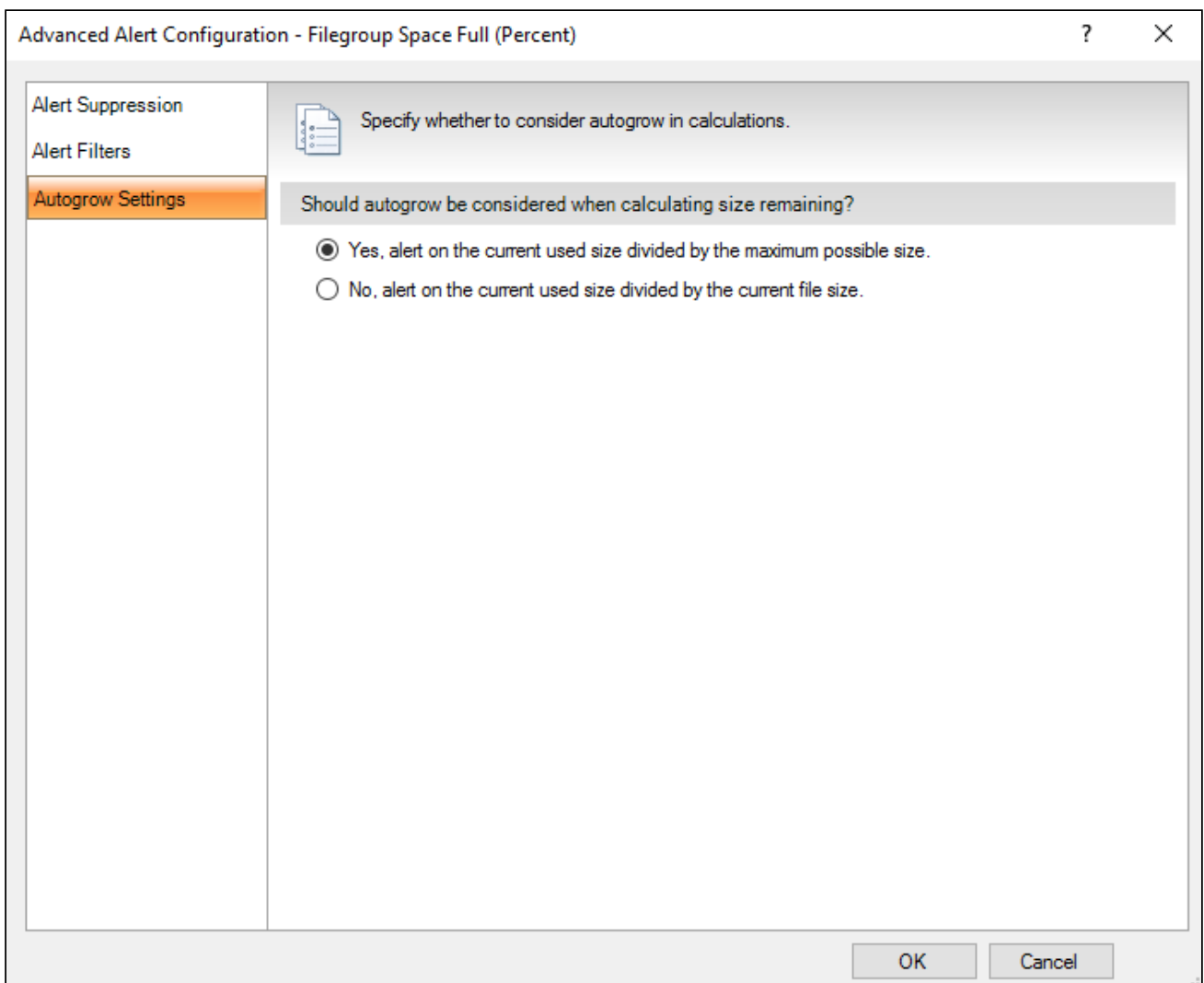
Filegroup Space Full (Percent) alert

The Filegroup Space Full (Percent) alert provides the percentage of the allowable filegroup space for the database currently used by all files within the filegroup. Allowable disk space for the database is calculated by taking into account the current allocated space, SQLdm auto-growth settings, and available disk space.

This alert is raised based on the database metrics collection interval specified in Server Properties. For more information, see [Set general server options](#).

Calculate the filegroup space full percentage using the autogrow settings

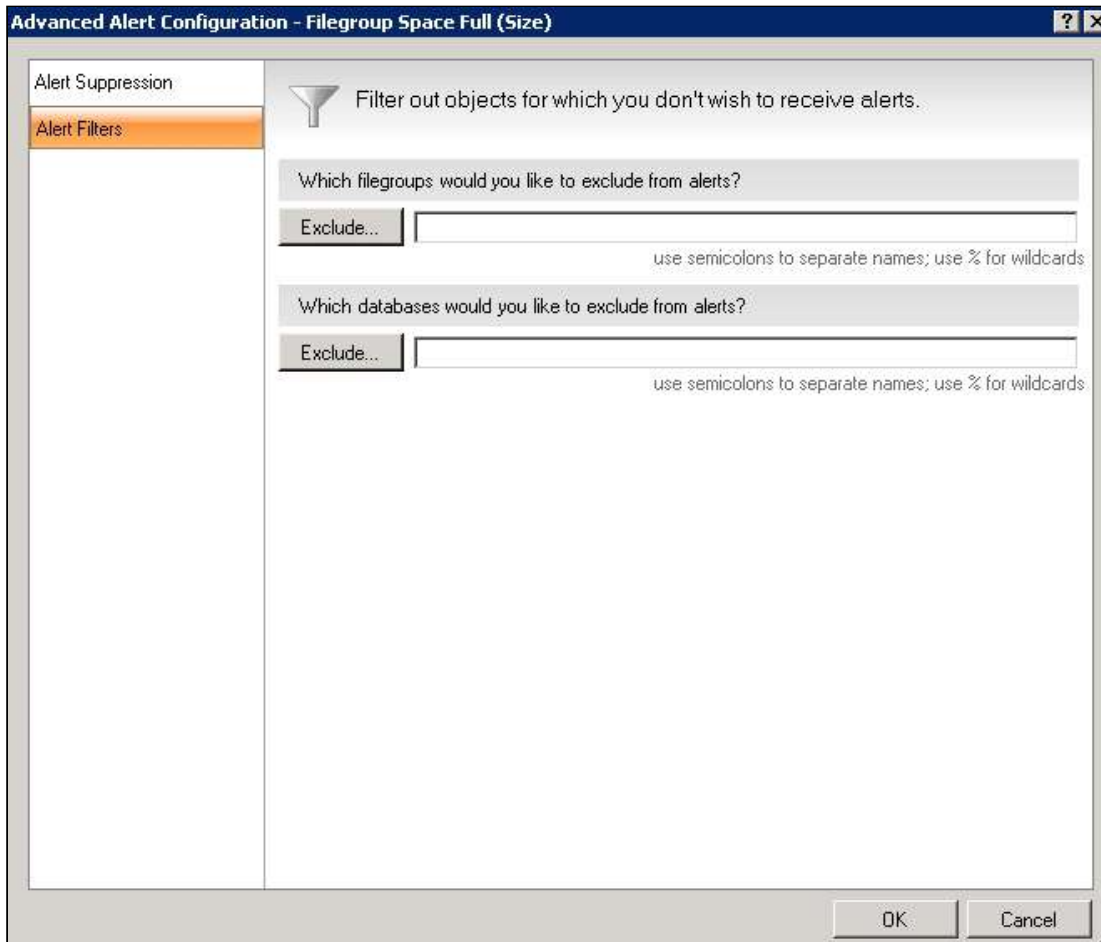
When setting the thresholds for your Filegroup Space Full (Percent) alert, click **Advanced > Autogrow Settings** (see image below) and SQL Diagnostic Manager provides a choice for you to calculate your displayed percentage. Select **Yes, alert on the current used size divided by the maximum possible size** to take possible autogrowth into consideration when calculating the filegroup space full percentage. Select **No, alert on the current used size divided by the current file size** to calculate the filegroup space full percentage based only on the current size of the database disk file.



Filegroup Space Full (Size) alert

The Filegroup Space Full (Size) alert provides the amount of disk space, in megabytes, currently used by the database as the sum of all the files within the filegroup.

SQL Diagnostic Manager raises this alert based on the database metrics collection interval specified in [Server Properties](#). You can also use advanced alert options such as filters to exclude specific filegroups or databases from alerting results.



You can access advanced filters by clicking **Advanced > Alert Filters** when setting the thresholds for your Filegroup Space Full (Size) alert.

⚠ The Filegroup Space Full (Size) alert is disabled by default.

Full-Text Search Status alert

The Full-Text Search Status alert provides the status of the Full-Text Search service.

Host CPU Usage (Percent) alert

The Host CPU Usage (Percent) alert indicates an unusually high amount of CPU usage by the VMWare or Hyper-V host server hosting the virtual machine on which the monitored SQL Server instance resides. The percentage of host server processor usage is listed under the control. A high host server percentage could indicate a large number of active client sessions. This alert is enabled by default and is available only on instances hosted on virtual machines enabled for VM monitoring.

Reduce your host CPU usage

Consistently high host server processor usage could indicate the need to:

- Reduce the number of SQL re-compilations as they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled and you can remove most of these reasons by careful coding.
- Make sure that all T-SQL statements (whether in a stored procedure, trigger or ad hoc statement) that reference objects fully qualify the object referenced.

For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. Both of these situations consume a significant amount of CPU time.

Host Memory Swap Detected alert

The Host Memory Swap Detected alert indicates that memory is swapped to disk for the host server hosting the virtual machine on which the monitored SQL Server resides. This may be an indication of memory pressure on the host. This alert is disabled by default and is only relevant to instances hosted on VMWare vCenter which have been enabled for VM monitoring in the SQLdm VM Configuration window.

Host Memory Usage (Percent) alert

The Host Memory Usage (Percent) alert provides the percent of available memory used by the host server hosting the virtual machine on which the monitored SQL Server resides. This alert is enabled by default and is available only on instances hosted on virtual machines enabled for VM monitoring.

Reduce your host memory usage

If this value is regularly over 80%, SQL Server needs more memory or needs to use the memory it has more efficiently. Consider implementing one or more of the following solutions:

- If your site makes use of extended stored procedures that are infrequently called, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never again called, which then wastes available memory. A `DBCC dllname (FREE)` releases that memory for use by both the procedure cache and Data Pages, which has a significant positive effect on both the Procedure Cache Hit Ratio and the Buffer Cache Hit Ratio. Execute `sp_helpextendedproc` to view the extended stored procedures currently loaded in memory by SQL Server.
- As each SQL Server lock requires 96 bytes of memory, the granting of lock space is done at the expense of Data Pages and Procedure Cache Pages. To maintain proper system performance and throughput, keep the number of locks to a minimum by:
 - Wherever possible, using the `(NOLOCK)` optimizer hint or `SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED` on select statements as this neither issues any shared locks on the data it reads nor honors any exclusive locks set by other transactions.
 - When updating all rows in a table with more than 50 rows, using the `TABLOCKX` table hint. This table hint prevents SQL Server from initially taking exclusive row locks, granting many of these locks, and then escalating them to an exclusive table lock.
 - When deleting all rows in any table, using the `TRUNCATE TABLE` statement instead of the `DELETE` statement as fewer locks and other system resources are consumed in the process.
 - Reducing the time that a lock is held by:
 - Performing as much work as possible before the transaction performs its first update, delete, or insert. For example, add any necessary `SELECT` statements.
 - Grouping all `UPDATES`, `DELETES`, and `INSERTS` as closely as possible within a transaction with as few `SELECTS` as possible separating them.
 - Committing the transaction as soon as possible after the final DML statement.
 - Avoiding any stops for user input once the transaction begins. Be sure to gather all user inputs before the transaction starts.
 - Allowing SQL Server to consume more of the available memory, making sure that OS Paging does not increase.
- Add more physical memory (RAM) to the computer.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.
- If the computer is running other memory-intensive applications, such as IIS or Exchange, then consider moving each instance to a separate physical computer.
- Limit SQL Server computers to performing only SQL Server work. Stop any unnecessary programs, such as allowing the computer to act as either a primary or backup domain controller.

Host Power State alert

The Host Power State alert provides the power state of the host as reported by the vCenter or Hyper-V host servers. This alert is enabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

Index Row Hits (Percent) alert

The Index Row Hits (Percent) alert provides the percentage of the density of an index as used by the SQL Query Optimizer to determine index effectiveness. A high density indicates the existence of many duplicates and an inefficient index. A low density indicates high selectivity and an effective index.

This alert is raised based on the database metrics collection interval specified in [Server Properties](#).

 This alert is for visual indicators in the **Desktop Client only**. SQL Diagnostic Manager does not send notification when this alert is triggered.

In-Memory Storage Usage Percent High

This metric is the percentage of storage used for in-memory transactions within the database, relative to the maximum available for the service tier.

In-Memory Storage Usage Percent Low

The In-Memory Storage Usage Percent Low is the percentage of storage used for in-memory transactions within the database, relative to the maximum available for the service tier. Consistently low values indicate that the service tier may be more than what is required for the workload.

Last Full-Text Catalog Update (Hours) alert

The Last Full-Text Catalog Update (Hours) alert provides the time, in hours, since the last Full-Text Catalog update was performed and beyond which the statistics data is considered outdated.

Since full-text indexes are contained in full-text catalogs, when you create new content or change existing content in a catalog, you must regenerate the full text index. This, along with [inefficient indexes](#) can impact your monitored SQL Server performance.

SQL Diagnostic Manager allows users to optimize full-text catalogs, for details see [Monitor Your Full-Text Searches](#).

Note that this alert is raised based on the diagnostic data collection interval specified in [Server Properties](#).

 This alert is for visual indicators in the **Desktop Client only**. SQL Diagnostic Manager does not send notification when this alert is triggered.

Latent replication transaction (Count)

The Latent replication transaction (Count) alert provides the number of replication transactions published by the Publisher but not received by the Distributor.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

Log File Autogrow alert

The Log File Autogrow alert indicates that a log file autogrow occurred on the specified database. For this alert to display, enable monitoring of non-query activities with capturing of autogrow events. Note that you can avoid unnecessarily long reads and poor response time from SQL Server by minimizing the number of Autogrow events occurring in your environment.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

To learn more about how autogrow works and related best-practices, see [Microsoft Support Article 315512](#), "Considerations for the 'autogrow' and 'autoshrink' settings in SQL Server".

Log Full (Percent) alert

The Log Full (Percent) alert provides the percentage of the allowable disk space for the database currently used by the transaction log. Allowable disk space for the database is calculated by taking into account the current allocated space, auto-growth settings and available disk space. If this value exceeds 80%, consider replacing disks with bigger disks.

This alert is raised based on the database metrics collection interval specified in Server Properties. For more information, see [Set general server options](#).

Calculate the log full percentage using the autogrow settings

When setting the thresholds for your Log Full (Percent) alert, click **Advanced > Autogrow Settings** and SQL Diagnostic Manager provides a choice for you to calculate your displayed percentage. Select **Yes, alert on the current used size divided by the maximum possible size** to take possible autogrowth into consideration when calculating the log full percentage. Select **No, alert on the current used size divided by the current file size** to calculate the log full percentage based only on the current size of the database disk file.

Log Full (Size) alert

The Log Full (Size) alert provides the amount of disk space, in megabytes, currently used by the transaction log for the database. This value represents the actual amount of data within the log and does not represent the size of the database log file itself.

This alert is raised based on the database metrics collection interval specified in Server Properties. For more information, see [Set general server options](#).

Longest Running Version Store Transaction (Minutes) alert

The Longest Running Version Store Transaction (Minutes) alert provides the time in minutes that the longest version store transaction is open and preventing version store cleanup. This alert occurs only on instances running SQL Server 2005 or above.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

Maintenance Mode Enabled alert

The Maintenance Mode Enabled alert indicates that the SQLDM Maintenance Mode is enabled. Maintenance Mode allows you to temporarily stop alert generation and the collection of performance metrics for the time period that this SQL Server instance is offline. You cannot configure this alert.

Max Session Percent High

The Max Session Percent High metric is the percentage of user sessions on the database, relative to the maximum available for the service tier. High values indicate that there may not be enough sessions available to support the users or processes.

Max Session Percent Low

The Max Session Percent Low metric is the percentage of user sessions on the database, relative to the maximum available for the service tier. Consistent low values indicate that the service tier may be more than what is required for the number of connections.

Max Worker Percent High

The Max Worker Percent High metric is the percentage of worker threads used on the database, relative to the maximum available for the service tier.

Max Worker Percent Low

The Max Worker Percent Low metric is the percentage of worker threads used on the database, relative to the maximum available for the service tier. Consistent low values indicate that the service tier may be more than what is required for the workload.

Mirror Commit Overhead (Milliseconds) alert

The Mirror Commit Overhead (Milliseconds) alert indicates that the commit overhead (in MS) of the mirrored database reached or exceeded an alert threshold. Specifies the number of milliseconds of average delay per transaction that are tolerated before a warning is generated on the principal server. This delay is the amount of overhead incurred while the principal server instance waits for the mirror server instance to write the transaction's log record into the redo queue. This value is relevant only in high-safety mode.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

For more information about mirror commit overhead, see the Microsoft document, [Use Warning Thresholds and Alerts on Mirroring Performance Metrics \(SQL Server\)](#).

Mirrored Server Role Change alert

The Mirrored Server Role Change alert indicates when the role of a mirrored database changes.

When you configure this alert, you can set the time at which the alert resets. Reset this alert when the role of the mirrored database has no impact.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

Mirroring Oldest Unsent Transactions (Minutes) alert

The Mirroring Oldest Unsent Transactions (Minutes) alert provides the age of unsent transactions queued on the processor and reached an alert threshold.

This alert specifies the number of minutes worth of transactions that can accumulate in the send queue before a warning is generated on the principal server instance. This warning helps measure the potential for data loss in terms of time and is especially relevant for high-performance mode. However, the warning is also relevant for high-safety mode when mirroring is paused or suspended because the partners become disconnected.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

For more information about oldest unsent transactions, see the Microsoft document, [Use Warning Thresholds and Alerts on Mirroring Performance Metrics \(SQL Server\)](#).

Mirroring Preferred Configuration alert

The Mirroring Preferred Configuration alert indicates that a database mirror partner took on a non-preferred role. This event means the roles in the mirroring relationship do not conform to the roles specified as Normal. This alert does not reset over time and remains until you restore the roles or you change the Normal state. You can restore the roles or modify the Normal definition in the **Mirroring** view of the Databases tab.

SQL Diagnostic Manager triggers this alert based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

Mirroring Status alert

The Mirroring Status alert provides the status of the mirrored database based on the current databases status other than OK. By default, the values Synchronizing and Synchronized equal an OK state. A Suspended database triggers a Warning alert, and a Disconnected or Pending Failover database status triggers the Critical alert.

You can change the status for the mirrored databases in the **Mirroring** view of the **Databases** tab.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

Mirroring Unrestored Log (KB) alert

The Mirroring Unrestored Log (KB) alert indicates that the amount of unrestored logs (in KB) queued on the mirrored database reached or exceeded an alert threshold.

This warning helps measure failover time. Failover time consists mainly of the time that the mirror server requires to roll forward any log remaining in its redo queue, plus a short additional time.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

For more information about unrestored logs, see the Microsoft document, [Use Warning Thresholds and Alerts on Mirroring Performance Metrics \(SQL Server\)](#).

Mirroring Unsent Log (KB) alert

The Mirroring Unsent Log (KB) alert indicates that the amount of unsent logs (in KB) queued on the mirrored database reached or exceeded an alert threshold.

This alert specifies how many kilobytes (KB) of unsent logs generate a warning on the principal server instance. This warning helps measure the potential for data loss in terms of KB and is especially relevant for high-performance mode. However, the warning is also relevant for high-safety mode when mirroring is paused or suspended because the partners become disconnected.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

For more information about unsent logs, see the Microsoft document, [Use Warning Thresholds and Alerts on Mirroring Performance Metrics \(SQL Server\)](#).

Mirroring Witness Connection alert

The Mirroring Witness Connection alert indicates that the connection to the mirror is lost.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

Oldest Open Transaction (Minutes) alert

The Oldest Open Transaction (Minutes) alert provides the time (in minutes) that the longest running transaction in the database has used since issuing the `BEGIN TRANSACTION` statement.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

OLE Automation Disabled alert

The OLE Automation Disabled alert indicates that the OLE automation configuration option is disabled in SQL Server 2005 or the procedures are dropped in SQL Server 2000 for the SQL Server instance. SQL Diagnostic Manager does not collect data such as OS metrics and service details using OLE automation. Use the OS Metrics tab of the server Properties dialog to reconfigure this option. For additional information about OS metrics collection, see [Configure OS metrics monitoring](#).

OS Average Disk Queue Length (Count) alert

The OS Average Disk Queue Length (Count) alert provides the average number of both read and write requests queued for all the disks on the SQL Server computer.

Reduce queue length

If this queue often remains above six per physical disk for more than 10 seconds at a time, then your disk subsystem is overloaded. In this case consider the following:

- Check OS Paging to make sure that paging from/to the swap file is not causing these IOs.
- Replacing disks with faster disks.
- Add more disks to you RAID array.
- Switch your RAID array from RAID 5 to RAID 10 solution as each write IO results in two writes using RAID 10 vs. 4 for RAID 5.
- Move other applications to another computer.
- So long as the RAID controller has some form of battery backup then switch its cache mode from Write-through to Write-back as this increases the system ability to handle write IOs by an order of magnitude.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.

✔ Enable baseline thresholds

To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

✔ Create an alert response bundle

Create an alert response bundle with the OS Average Disk Queue Length (Count) alert and related alerts. For additional information, see [Create alert response bundles](#).

OS Average Disk Queue Length Per Disk (Count) alert

The OS Average Disk Queue Length Per Disk (Count) alert provides the average number of both read and write requests queued for each disk on the SQL Server computer.

Reduce queue length

If this queue often remains above six per physical disk for more than 10 seconds at a time then your disk subsystem is overloaded. In this case consider the following:

- Check OS Paging to make sure that paging from/to the swap file is not causing these IOs.
- Replacing disks with faster disks.
- Add more disks to you RAID array.
- Switch your RAID array from RAID 5 to RAID 10 solution as each write IO results in two writes using RAID 10 vs. 4 for RAID 5.
- Move other applications to another computer.
- So long as the RAID controller has some form of battery backup then switch its cache mode from Write-through to Write-back as this increases the system ability to handle write IOs by an order of magnitude.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.



Create an alert response bundle with the OS Average Disk Queue Length Per Disk (Count) alert and related alerts. For additional information, see [Create alert response bundles](#).

OS Disk Free Space (Size) alert

The OS Disk Free Space (Size) alert provides the amount of available disk space, in megabytes, on the SQL Server computer. This alert triggers separately for each disk on a server.

- ✔ Create an alert response bundle with the OS Disk Free Space (Size) alert and related alerts. For additional information, see [Create alert response bundles](#).

OS Disk Full (Percent) alert

The OS Disk Full (Percent) alert provides the percentage of space used on a logical disk on the SQL Server computer. This alert is triggered separately for each disk on a server.

- ✓ You can access the Disk Size view of the Resources tab by double-clicking the OS Disk Full (Percent) alert in the Alerts view.

- ✓ **Create an alert response bundle**

Create an alert response bundle with the OS Disk Full (Percent) alert and related alerts. For additional information, see [Create alert response bundles](#).

OS Disk Time (Percent) alert

The OS Disk Time (Percent) alert provides the percentage of elapsed time that all disks are busy servicing read and write requests on the SQL Server computer.

Reduce OS processor time

If this value regularly exceeds 75%, you should take action such as:

- Reduce the number of SQL re-compilations since they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled and you can remove most of these reasons by careful coding. See the SQL Re-compilations counter for ways to dramatically reduce recompiles.
- Make sure that all T-SQL statements (whether in a stored procedure, trigger, or ad hoc statement) that reference objects fully qualify the object referenced. For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. Both of these situations consume a significant amount of CPU time.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.



Enable baseline thresholds

To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.



Create an alert response bundle

Create an alert response bundle with the OS Disk Time (Percent) alert and related alerts. For additional information, see [Create alert response bundles](#).

OS Disk Time Per Disk (Percent) alert

The OS Disk Time Per Disk (Percent) alert provides the percentage of elapsed time that an individual disk is busy servicing read and write requests on the SQL Server computer. This alert triggers separately for each disk on a server.

Reduce OS processor time

If this value regularly exceeds 75%, you should take action such as:

- Reduce the number of SQL re-compilations since they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled and you can remove most of these reasons by careful coding. See the SQL Re-compilations counter for ways to dramatically reduce recompiles.
- Make sure that all T-SQL statements (whether in a stored procedure, trigger, or ad hoc statement) that reference objects fully qualify the object referenced. For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. Both of these situations consume a significant amount of CPU time.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.



Create an alert response bundle with the OS Disk Time Per Disk (Percent) alert and related alerts. For additional information, see [Create alert response bundles](#).

OS Memory Usage (Percent) alert

The OS Memory Usage (Percent) alert provides the percentage of physical and virtual memory. Consistently high Total Memory Usage could indicate either:

- A memory leak in a program in use.
- A need to increase the amount of total memory.

The Memory view on the Resources tab allows you to track the Total Memory usage along with other key memory statistics on the computer hosting your SQL Server instance.

Reduce OS memory usage

A high percentage of Total Memory Usage indicates a memory bottleneck. To investigate this issue, open the Memory view on the Resources tab to view how memory is utilized on your SQL Server instance. You may need to allocate more memory to SQL Server to increase your SQL Server performance, or limit the usage of other applications on the computer hosting your SQL Server instance.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

OS Metrics Collection Status alert

The OS Metrics Collection Status alert provides the status of SQL diagnostic manager OS Metrics Collection.

OS Paging (Per Second) alert

The OS Paging (Per Second) alert indicates that a memory threshold meets or exceeds the setting, due to one of the following causes:

- The OS Memory Paging is high
- The OS Memory Usage is high
- The SQL Server Memory Usage is high

Consistently high memory usage could cause system performance problems because of heavy disk usage and CPU load.

Reduce memory paging

A high number of memory paging indicates a memory bottleneck. You should open the **Memory** view on the **Resources** tab to see how memory is being utilized on your SQL Server instance. You may need to allocate more memory to SQL Server to increase your SQL Server performance.

- ✓ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

OS Privileged Time (Percent) alert

The OS Privileged Time (Percent) alert provides the percentage of CPU time spent by all processes executing in privileged mode on the SQL Server computer.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

OS Processor Queue Length (Count) alert

The OS Processor Queue Length (Count) alert indicates that the number of threads waiting on the processor reached an alert threshold. A consistently high processor queue length can indicate processor congestion. If your system has multiple processors, you may need to increase your alert thresholds.

Reduce your OS processor queue length

If the Processor Queue Length is often high, consider making one of the following changes:

- Upgrade to faster CPU(s).
- Reduce the number of SQL re-compilations as they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled, and you can remove most of these reasons by careful coding. See the SQL re-compilations counter for ways to dramatically reduce recompiles.
- Stop unnecessary programs from running (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC).



To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

OS Processor Time (Percent) alert

The OS Processor Time (Percent) alert provides the percentage of CPU time used by all processes on the computer hosting your SQL Server instance. The WMI object `Win32_PerfRawData_PerfOS_Processor` property `PercentProcessorTime` is measured to get percent processor time.

Reduce OS processor time

If this value regularly exceeds 75%, you should take action such as:

- Reduce the number of SQL re-compilations since they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled and you can remove most of these reasons by careful coding. See the SQL Re-compilations counter for ways to dramatically reduce recompiles.
- Make sure that all T-SQL statements (whether in a stored procedure, trigger, or ad hoc statement) that reference objects fully qualify the object referenced. For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. Both of these situations consume a significant amount of CPU time.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.

Reasons why the CPU Usage metric has no value

SQL Diagnostic Manager is unable to collect the corresponding counter due to one of the following reasons:

- It usually takes two refreshes to begin populating the CPU Usage metric, which occurs within a few seconds.
- OS Metrics collection is disabled or not working properly. You must enable OS Metrics collection for SQL Diagnostic Manager to collect hardware performance metrics.
- Lightweight Pooling is enabled. SQL Diagnostic Manager cannot collect OS metrics if Lightweight Pooling is enabled on the SQL Server instance.
- Alerts are disabled. SQL Diagnostic Manager displays an alert status when at least one of the three alerts is enabled.
- The counter does not exist on this SQL Server instance.



To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

OS User Time (Percent) alert

The OS User Time (Percent) alert provides the percentage of CPU time spent by all processes executing in user mode on the SQL Server computer.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

Page Life Expectancy alert

The Page Life Expectancy alert is triggered when the number of seconds a page stays in the buffer pool without references is below the warning level. A decrease in page life expectancy can indicate an increase in the physical I/O requirements for a user database. The rate decrease could likely indicate that the memory taken away from the buffer pool is forcing database pages to exit the buffer pool prematurely.

Diagnose problems with your SQL Server instance

Microsoft recommends a minimum target for page life expectancy of 300 seconds. This means that any given page in memory is kept in the buffer for five minutes before the buffer flushes the page to disk. A page life expectancy value of less than 300 seconds is indicative of either a memory problem or inefficient query plans.

- ✔ To enable alerting when this metric is outside its established baseline, select the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

Procedure Cache Hit Ratio (Percent) alert

The Procedure Cache Hit Ratio (Percent) alert provides the ratio of procedure cache hits to procedure cache lookups. This value indicates the reuse of an execution plan from memory as opposed to a compilation from disk.

Manage your Procedure Cache Hit Ratio

If you regularly encounter values below 80% once SQL Server is actively running for at least 30 minutes, then consider one of the following remedies:

- Make sure that all T-SQL statements (whether in a stored procedure, trigger or ad hoc statement) that reference objects fully qualify the object referenced. For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. Both of these situations consume a significant amount of CPU time.
- Add more physical memory (RAM) to the computer.
- If your site makes use of extended stored procedures that are not called very often, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never called again! This is tremendously wasteful in terms of available memory and a `DBCC DLLNAME (FREE)` releases that memory for use by both the procedure cache as well as the Data Pages which has a significant positive effect on both the Procedure Cache Hit Ratio, as well as the Buffer Cache Hit Ratio. In order to view the extended stored procedures currently loaded in memory by SQL Server, execute `SP_HELPXTEENDEDPROC`.
- Allow SQL Server to consume more of the available memory (making sure that OS Paging does not increase).
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.
- If the computer is running multiple instances of SQL Server, consider placing each instance on a separate physical computer.



To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

Query Monitor Events (Count) alert

The Query Monitor Events (Count) alert provides the number of queries captured by the Query Monitor on the SQL Server instance during the last scheduled refresh. A warning is generated if a query matches the criteria specified for a poorly performing query.

Read Latency

The Read Latency metric is the measure of how long it takes for the SQL Server instance to complete a read from storage.

Read Throughput

The Read Throughput metric is the measure of how many MB/sec is being read from storage.

Service Tier Changes

The Service Tier Changes metric indicated whether the service level for a database has been changed in the time period being checked.

Session CPU Time (Seconds) alert

The Session CPU Time (Seconds) alert provides the CPU Time (in seconds) used by the SQL Server session since the last scheduled refresh.

Session Tempdb Space Usage (MB) alert


The Session Tempdb Space Usage (MB) alert provides the amount of tempdb space being used by a running session. This alert occurs only on instances running SQL Server 2005 or above.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

SQLdm Repository Grooming Timed Out alert

The 'SQLdm Repository Grooming Timed Out' alert is triggered when the grooming process of your SQLDM Repository has reached the time out period (180 minutes).

The time out period is typically exceeded when the repository database becomes very large in size. Running a complete reindex of the database should help improve the performance of the grooming process and prevent SQL DM from raising this alert.

 To review your grooming settings, go to **Tools > Grooming Options**.

SQLDM Service Status alert

The SQLDM Service Status alert provides the status of the SQLDM Manager Service. The possible statuses include **Running**, **Stopped**, or **Unknown**. You cannot configure this alert.

SQL Server Active Directory Helper Service Status alert

This alert provides the status of the SQL Server Active Directory Helper service.

SQL Server Agent Job Completion alert

The SQL Server Agent Job Completion alert provides the status of the Agent Job Completion metric. Values for this alert include **Succeeded**, **Retry**, **Canceled**, **Failed**, and **Unknown**.

- ✔ Create an alert response bundle with the SQL Server Agent Job Completion alert and related alerts. For additional information, see [Create alert response bundles](#).

SQL Server Agent Job Failure alert

The SQL Server Agent Job Failure alert indicates that the scheduled SQL Server Agent job ended with a Failed status.

- ✔ Create an alert response bundle with the SQL Server Agent Job Failure alert and related alerts. For additional information, see [Create alert response bundles](#).

SQL Server Agent Log alert

The SQL Server Agent Log alert indicates that specific text appeared in the SQL Server Agent error log. You can customize which text triggers this alert by using the Alert Filters tab on the Advanced Alert Configuration window.

The default settings are:

- 2 - Warning
- 1 - Critical

Use a size limit on your SQL Server Agent log

SQL Diagnostic Manager allows you to limit the size of your SQL Server Agent log to help improve response time and avoid using unnecessary space on your server. For additional information and steps on setting this limit, see [Configure text and expression alerts](#).

- ✔ Create an alert response bundle with the SQL Server Agent log alert and related alerts. For additional information, see [Create alert response bundles](#).

SQL Server Agent Long Running Job (Minutes) alert

The SQL Server Agent Long Running Job (Minutes) alert provides the maximum number of minutes used to complete a scheduled SQL Server Agent job.

- ✔ Create an alert response bundle with the SQL Server Agent Long Running Job (Minutes) alert and related alerts. For additional information, see [Create alert response bundles](#).

SQL Server Agent Long Running Job (Percent) alert

The SQL Server Agent Long Running Job (Percent) alert provides the percentage of time that a scheduled SQL Server Agent job uses over the average successful completion time for this job in the current job history.

- ✔ Create an alert response bundle with the SQL Server Agent Long Running Job (Percent) alert and related alerts. For additional information, see [Create alert response bundles](#).

SQL Server Agent Status alert

The SQL Server Agent Status alert provides the status of the SQL Server Agent Service.

SQL Server Agent XPs Disabled alert

The SQL Server Agent XPs Disabled alert indicates that the Agent XPs configuration option is disabled in SQL Server 2005 or later versions. The alert may also signal that procedures are dropped in SQL Server 2000 for the SQL Server instance. Note that SQL Diagnostic Manager does not collect SQL Agent job details.

Agent XPs are disabled when SQL Server Agent is stopped. You can re-enable Agent XPs through the Server Configuration view and allow viewing of agent job histories even when the agent is not running.

SQL Server Browser Service Status alert

This alert provides the status of the SQL Server Browser service.

SQL Server CPU Usage (Percent) alert

The SQL Server CPU Usage (Percent) alert indicates an unusually high amount of CPU usage by SQL Server. The percentage of SQL Server processor usage is listed under the control. A high SQL Server percentage could indicate a large number of active client sessions.

Reduce your SQL Server CPU usage

Consistently high SQL Server processor usage could indicate the need to:

- Reduce the number of SQL re-compilations as they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled and you can remove most of these reasons by careful coding.
- Make sure that all T-SQL statements (whether in a stored procedure, trigger or ad hoc statement) that reference objects fully qualify the object referenced.

For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. Both of these situations consume a significant amount of CPU time.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

SQL Server Data Used (Percent) alert

The SQL Server Data Used (Percent) alert indicates that you have a high percentage of allocated database space in use.

This alert is raised based on the database metrics collection interval specified in Server Properties. For more information, see [Set general server options](#).

Reduce your SQL Server database space used

- You may need to increase the amount of space allocated to databases, add disks to the computer hosting the instance, or move databases to another host computer.
- If you have autogrowth turned off, this alert indicates that you need to either allocate more space for databases or add more disk space.
- If you have autogrowth turned on and this alert is triggered often, you may need to adjust your autogrowth settings to a higher percentage to improve performance. Constant growth adjustments can degrade performance.
- For SQL Server 2000 users, SQLdm may incorrectly report the file size. This issue occurs when certain columns in the `sysindexes` table become inaccurate and provide SQL Diagnostic Manager with incorrect information regarding the number of rows and table size. To correct this problem, use the following T-SQL syntax:

```
DBCC UPDATEUSAGE ('database_name' ) GO
```

We recommend that you execute this command during non-peak hours as it is server intensive.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

SQL Server Error Log alert

The SQL Server Error Log alert indicates that specific text appeared in the SQL Server log. You can customize which text triggers the alert.

The default settings are:

- 11 - Warning
- 18 - Critical

Use a size limit on your SQL Server Error log

SQL Diagnostic Manager allows you to limit the size of your SQL Server Error log to help improve response time and avoid using unnecessary space on your server. For additional information and steps on setting this limit, see [Configure text and expression alerts](#).

- ✔ Create an alert response bundle with the SQL Server Error Log alert and related alerts. For additional information, see [Create alert response bundles](#).

SQL Server Log Used (Percent) alert

The SQL Server Log Used (Percent) alert provides the percentage of allocated disk space used by logs on the SQL Server instance reached an alert threshold. Once the log space is completely filled, all updates to the event log fail.

This alert is raised based on the database metrics collection interval specified in Server Properties. For more information, see [Set general server options](#).

Reduce your SQL Server log space used

- You can allocate more disk space to the event log, turn on the auto-growth feature for your SQL Server log, or add additional disk space to the computer hosting your SQL Server instance.
- If autogrowth is turned off for your logs, this alert may indicate the need to either increase the allocated space for logs and/or add disk space to your SQL Server environment.
- If autogrowth is turned on and you are seeing this alert often, you may need to adjust your autogrowth settings to a higher percentage to improve degraded performance resulting from constant growth adjustments.
- In certain cases, SQL Diagnostic Manager may incorrectly report the file size. This issue occurs when certain columns in the 'sysindexes' table in SQL Server 2000 become inaccurate and provide SQL Diagnostic Manager with incorrect information regarding the number of rows and table size.
- Use the following T-SQL syntax to correct this issue:

```
DBCC UPDATEUSAGE ( 'database_name' ) GO
```

We recommend that you execute this command during non-peak hours as it is server intensive.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

SQL Server Memory Usage (Percent) alert

The SQL Server Memory Usage (Percent) alert provides the total server memory (Total Server Memory per `sysperfinfo`) as a percent of total physical memory from WMI.

Reduce the percentage of memory used by SQL Server

If this value is regularly over 80%, SQL Server needs more memory or needs to use the memory it has more efficiently. Consider implementing one or more of the following solutions:

- If your site makes use of extended stored procedures that are infrequently called, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never again called, which then wastes available memory. A `DBCC dllname (FREE)` releases that memory for use by both the procedure cache and Data Pages, which has a significant positive effect on both the Procedure Cache Hit Ratio and the Buffer Cache Hit Ratio. Execute `sp_helpextendedproc` to view the extended stored procedures currently loaded in memory by SQL Server.
- As each SQL Server lock requires 96 bytes of memory, the granting of lock space is done at the expense of Data Pages and Procedure Cache Pages. To maintain proper system performance and throughput, keep the number of locks to a minimum by:
 - Wherever possible, using the `(NOLOCK)` optimizer hint or `SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED` on select statements as this neither issues any shared locks on the data it reads nor honors any exclusive locks set by other transactions.
 - When updating all rows in a table with more than 50 rows, using the `TABLOCKX` table hint. This table hint prevents SQL Server from initially taking exclusive row locks, granting many of these locks, and then escalating them to an exclusive table lock.
 - When deleting all rows in any table, using the `TRUNCATE TABLE` statement instead of the `DELETE` statement as fewer locks and other system resources are consumed in the process.
 - Reducing the time that a lock is held by:
 - Performing as much work as possible before the transaction performs its first update, delete, or insert. For example, add any necessary `SELECT` statements.
 - Grouping all `UPDATES`, `DELETES`, and `INSERTS` as closely as possible within a transaction with as few `SELECTS` as possible separating them.
 - Committing the transaction as soon as possible after the final DML statement.
 - Avoiding any stops for user input once the transaction begins. Be sure to gather all user inputs before the transaction starts.
 - Allowing SQL Server to consume more of the available memory, making sure that OS Paging does not increase.
- Add more physical memory (RAM) to the computer.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.
- If the computer is running other memory-intensive applications, such as IIS or Exchange, then consider moving each instance to a separate physical computer.
- Limit SQL Server computers to performing only SQL Server work. Stop any unnecessary programs, such as allowing the computer to act as either a primary or backup domain controller.



To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

SQL Server Response Time (Milliseconds) alert

The SQL Server Response Time (Millisecond) alert provides the time (in milliseconds) SQL Diagnostic Manager currently needs to send a simple SQL command to the SQL Server instance, have it processed, and receive the returned result set.

High response times generally occur when sessions are blocked or when either there is heavy network traffic or network problems or when either the computer the SQL Diagnostic Manager Repository or the computer the SQL Diagnostic Manager Console is hosted on is running slow due to various problems.

Typically, this value should average below 2000 ms for a well tuned network and server. Average values in excess of 2000 ms indicate either an excessively busy network segment or a stressed SQL Server.

Reduce your SQL Server response time

A common cause to high response times is when sessions are being blocked. To fix this problem, go to the **Sessions** tab and select the **Blocking** view to see if any sessions are blocking. You can then select the session and click either **Trace Session** to see what queries the session is creating and to see what is causing the block, or **Kill Session** to stop it.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

SQL Server Service

The SQL Server Status alert provides the status of the SQL Server service.

Swap Usage

The Swap Usage metric is used when there is not enough available memory to fulfill a request.

Table Fragmentation (Percent) alert

The Table Fragmentation (Percent) alert provides the percentage of fragmented pages in a table. It is the scan density from `dbcc showcontig` on a clustered index of a table. If the table does not have a clustered index, table fragmentation is not collected. If this value exceeds 80%, consider the following remedies:

- Drop and re-create the clustered index.
- Reorder the leaf-level pages of the index in a logical order.
- Rebuild the index online.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

Tempdb Contention (ms) alert

The Tempdb Contention (ms) alert provides the latch wait time (in milliseconds) for tempdb allocation maps. The three tracked allocation page types are:

- Page free space (PFS)
- Global allocation map (GAM)
- Shared global allocation map (SGAM)

Latch contention of this sort is usually an indication that you need to create more tempdb data files. In some situation, using Trace Flag 1118 may also alleviate tempdb contention. This alert occurs only on instances running SQL Server 2005 or above.

Unsubscribed Transactions (Count) alert

The Unsubscribed Transactions (Count) alert provides the number of replication transactions received by the Distributor but not received by a Subscriber.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

Unsubscribed Transactions (Seconds) alert

The Unsubscribed Transactions (Seconds) alert provides the time (in seconds) that a replication transaction is holding at the Distributor and waiting for a Subscriber to receive it.

- ✔ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.


User Connections (Percent) alert

The User Connections (Percent) alert provides the percentage of currently-used user connections allowed by the SQL Server instance. A high percentage indicates a load on the SQL Server. Note that the higher this value, the more SQL Server reserves for connections.

- ✓ To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

Version Store Generation Ratio alert

The Version Store Generation Ratio alert provides the percentage by which the tempdb version store generation rate exceeds the cleanup rate. This alert is disabled by default and occurs only on instances running SQL Server 2005 or above.

 You can raise this alert only when at least one minute elapses between alert refreshes as version store cleanup occurs once per minute.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

Version Store Size (MB) alert

The Version Store Size (MB) alert provides the size of the tempdb version store in megabytes. This alert is disabled by default and occurs only on instances running SQL Server 2005 or above.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

VM CPU Ready Wait Time (ms) alert

The VM CPU Ready Wait Time (MS) alert provides the time (in milliseconds) that a virtual CPU is waiting for a physical CPU to become available for processing for the virtual machine on which the monitored SQL Server resides. A spike in this value may indicate CPU pressure on the host server. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLDM VM Configuration window.

VM CPU Usage (Percent) alert

The VM CPU Usage (Percent) alert provides the percentage of available virtual CPU time used by the virtual machine hosting the monitored SQL Server. This may differ from the processor utilization percentage reported by the guest OS. The percentage of processor usage is listed under the control. A high percentage could indicate a large number of active client sessions. This alert is disabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

Reduce the percentage of available virtual CPU time used by your VM

Consistently high SQL Server processor usage could indicate the need to:

- Reduce the number of SQL re-compilations as they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled and you can remove most of these reasons by careful coding.
- Make sure that all T-SQL statements (whether in a stored procedure, trigger or ad hoc statement) that reference objects fully qualify the object referenced.

For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. Both of these situations consume a significant amount of CPU time.



To enable alerting when this metric is outside its established baseline, click the **Baseline Thresholds Enabled (as percentage of baseline)** check box in the Alert Configuration window.

VM Host Server Change alert

The VM Host Server Change alert indicates that the virtual machine hosting the monitored SQL Server instance moved to a different host. This alert is disabled by default and is only relevant to instances hosted on VMWare vCenter which have been enabled for VM monitoring in the SQLDM VM Configuration window.

VM Memory Swap Delay Detected alert

The VM Memory Swap Delay Detected alert indicates that the memory being swapped to disk is causing a performance delay on the virtual machine on which the monitored SQL Server instance is running. This alert is enabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

VM Memory Usage (Percent) alert

The VM Memory Usage (Percent) alert provides the percentage of available memory used by the virtual machine hosting the monitored SQL Server. This result may vary from the memory usage percentage reported by the guest operating system. This alert is disabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

Reduce the percentage of memory used by SQL Server

If this value is regularly over 80%, SQL Server needs more memory or needs to use the memory it has more efficiently. Consider implementing one or more of the following solutions:

- If your site makes use of extended stored procedures that are infrequently called, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never again called, which then wastes available memory. A `DBCC dllname (FREE)` releases that memory for use by both the procedure cache and Data Pages, which has a significant positive effect on both the Procedure Cache Hit Ratio and the Buffer Cache Hit Ratio. Execute `sp_helpextendedproc` to view the extended stored procedures currently loaded in memory by SQL Server.
- As each SQL Server lock requires 96 bytes of memory, the granting of lock space is done at the expense of Data Pages and Procedure Cache Pages. To maintain proper system performance and throughput, keep the number of locks to a minimum by:
 - Wherever possible, using the `(NOLOCK)` optimizer hint or `SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED` on select statements as this neither issues any shared locks on the data it reads nor honors any exclusive locks set by other transactions.
 - When updating all rows in a table with more than 50 rows, using the `TABLOCKX` table hint. This table hint prevents SQL Server from initially taking exclusive row locks, granting many of these locks, and then escalating them to an exclusive table lock.
 - When deleting all rows in any table, using the `TRUNCATE TABLE` statement instead of the `DELETE` statement as fewer locks and other system resources are consumed in the process.
 - Reducing the time that a lock is held by:
 - Performing as much work as possible before the transaction performs its first update, delete, or insert. For example, add any necessary `SELECT` statements.
 - Grouping all `UPDATES`, `DELETES`, and `INSERTS` as closely as possible within a transaction with as few `SELECTS` as possible separating them.
 - Committing the transaction as soon as possible after the final DML statement.
 - Avoiding any stops for user input once the transaction begins. Be sure to gather all user inputs before the transaction starts.
 - Allowing SQL Server to consume more of the available memory, making sure that OS Paging does not increase.
- Add more physical memory (RAM) to the computer.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.
- If the computer is running other memory-intensive applications, such as IIS or Exchange, then consider moving each instance to a separate physical computer.
- Limit SQL Server computers to performing only SQL Server work. Stop any unnecessary programs, such as allowing the computer to act as either a primary or backup domain controller.

VM Power State alert

The VM Power State alert provides the power state of the VM as reported by the vCenter or Hyper-V host servers. This alert is enabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

VM Reclaimed/Ballooned Memory (KB) alert

The VM Reclaimed/Ballooned Memory (KB) alert provides the amount of memory that is reclaimed by the host server and is no longer available to the virtual machine on which the SQL Server instance is running. This may be a sign of memory pressure on the host. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLDm VM Configuration window.

Reduce the percentage of memory used by SQL Server

If this value is regularly over 80%, SQL Server needs more memory or needs to use the memory it has more efficiently. Consider implementing one or more of the following solutions:

- If your site makes use of extended stored procedures that are infrequently called, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never again called, which then wastes available memory. A `DBCC dllname (FREE)` releases that memory for use by both the procedure cache and Data Pages, which has a significant positive effect on both the Procedure Cache Hit Ratio and the Buffer Cache Hit Ratio. Execute `sp_helpextendedproc` to view the extended stored procedures currently loaded in memory by SQL Server.
- As each SQL Server lock requires 96 bytes of memory, the granting of lock space is done at the expense of Data Pages and Procedure Cache Pages. To maintain proper system performance and throughput, keep the number of locks to a minimum by:
 - Wherever possible, using the `(NOLOCK)` optimizer hint or `SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED` on select statements as this neither issues any shared locks on the data it reads nor honors any exclusive locks set by other transactions.
 - When updating all rows in a table with more than 50 rows, using the `TABLOCKX` table hint. This table hint prevents SQL Server from initially taking exclusive row locks, granting many of these locks, and then escalating them to an exclusive table lock.
 - When deleting all rows in any table, using the `TRUNCATE TABLE` statement instead of the `DELETE` statement as fewer locks and other system resources are consumed in the process.
 - Reducing the time that a lock is held by:
 - Performing as much work as possible before the transaction performs its first update, delete, or insert. For example, add any necessary `SELECT` statements.
 - Grouping all `UPDATES`, `DELETES`, and `INSERTS` as closely as possible within a transaction with as few `SELECTS` as possible separating them.
 - Committing the transaction as soon as possible after the final DML statement.
 - Avoiding any stops for user input once the transaction begins. Be sure to gather all user inputs before the transaction starts.
 - Allowing SQL Server to consume more of the available memory, making sure that OS Paging does not increase.
- Add more physical memory (RAM) to the computer.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.
- If the computer is running other memory-intensive applications, such as IIS or Exchange, then consider moving each instance to a separate physical computer.
- Limit SQL Server computers to performing only SQL Server work. Stop any unnecessary programs, such as allowing the computer to act as either a primary or backup domain controller.

VM Resource Configuration Change alert

The VM Resource Configuration Change alert indicates that a resource configuration change occurred for the virtual machine hosting the monitored SQL Server instance. This alert is enabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

VM Resource Limits Detected alert

The VM Resource Limits Detected alert indicates that CPU or memory resource limits are set for the virtual machine hosting the monitored SQL Server instance. Resource limits are set by the VM administrator on the total amount of resources that a VM is allowed to consume. This alert is disabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

WMI Service Unavailable alert

The WMI Service Unavailable alert indicates that the WMI service is unavailable on the SQL Server computer. SQL Diagnostic Manager does not collect OS metrics.

Write Latency

Write Latency is the measure of how long it takes for the SQL Server instance to complete a write to storage.

Write Latency Low

Write latency is the measure of how long it takes for the SQL Server instance to complete a read from storage. High average write latency indicates that the storage system is not performing well or is overwhelmed.

Write Throughput

Write Throughput is a measure of how many MB/sec is being write to storage.


Advanced solutions

SQL Diagnostic Manager includes functionality aimed at the more advanced user who wants to manage these features without contacting IDERA Support for assistance. Please do not attempt to perform these actions if you are unsure of what you are doing or why you would want to take such steps. These advanced topics include the following actions:

- [Migrate or recover your installation](#)
- [Deploy the Idera Newsfeed in a clustered environment](#)
- [Deploy SQL Diagnostic Manager in a clustered environment](#)
- [Understand default alert IDs](#)
- [Troubleshoot WMI connectivity issues](#)
- [Integrate SQL Diagnostic Manager with SCOM](#)
- [Certificate Issues](#)

Migrate or Recover Your Installation

Establishing a migration or disaster recovery strategy for your SQL Diagnostic Manager Repository and Services allows you to preserve historical data and existing configuration settings. You can also continue monitoring your SQL Server environment to meet your auditing requirements.

 [SQL Diagnostic Manager Application security](#) is not maintained when your SQL Diagnostic Manager installation is migrated to a new server.

This process helps you execute a migration or disaster recovery strategy that addresses one of the following situations:

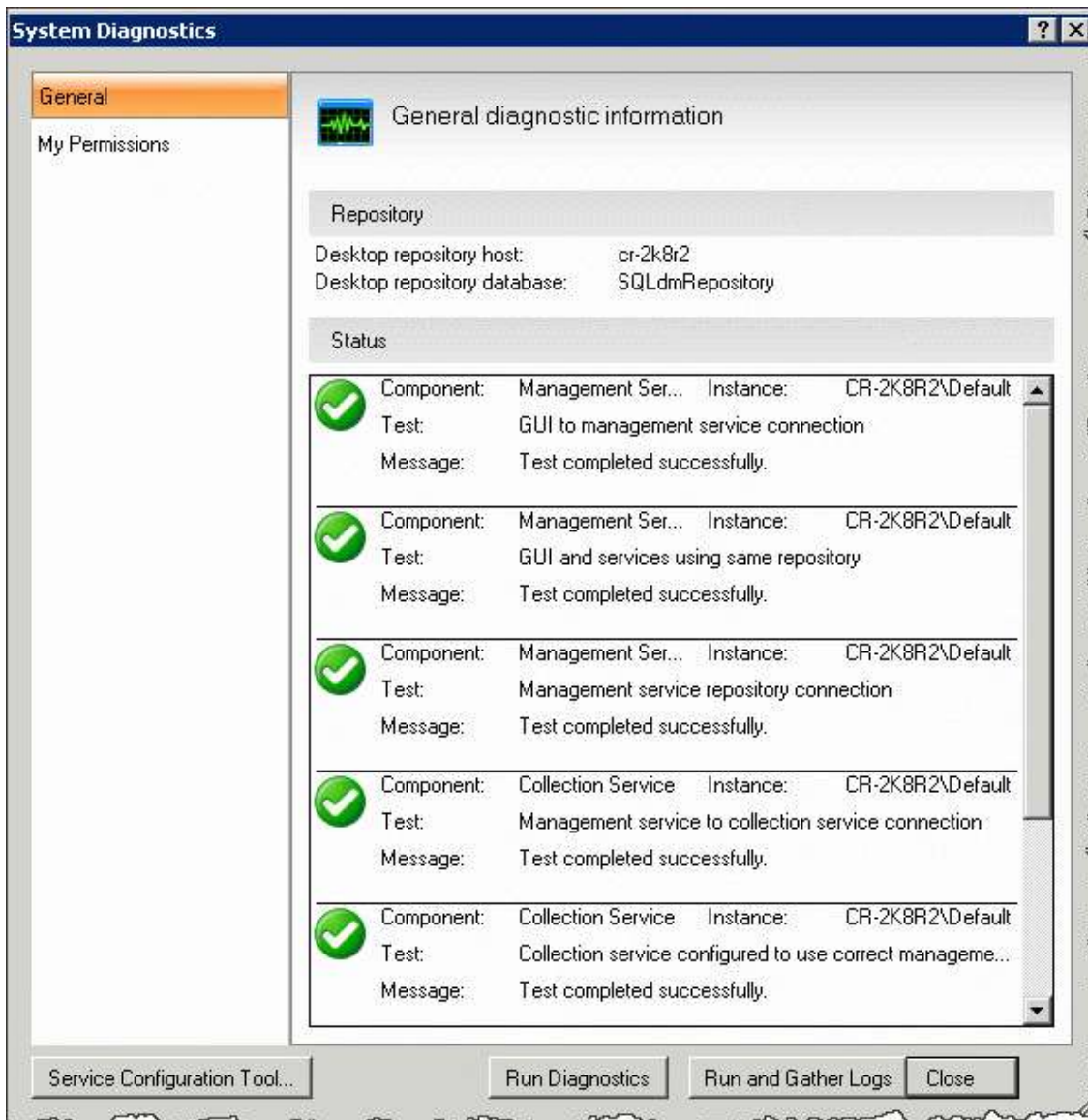
- The computer hosting SQL Diagnostic Manager requires maintenance, such as new hardware or a software upgrade (Microsoft Windows or SQL Server Service Pack)
- The computer hosting SQL Diagnostic Manager becomes permanently unavailable
- You decommission and replace the computer hosting SQL Diagnostic Manager

Refer to the following when migrating your SQL Diagnostic Manager installation:

- [Choosing to migrate or recover SQL Diagnostic Manager](#)
- [Creating your migration or recovery plan](#)
- [Recovering your SQL Diagnostic Manager installation](#)
- [Migrating your complete SQL Diagnostic Manager installation](#)
- [Migrating the SQL Diagnostic Manager Repository only](#)
- [Migrating the SQL Diagnostic Manager services only](#)

System diagnostics

The System Diagnostics window allows you to test the connections between the SQL Diagnostic Manager services and other SQL Diagnostic Manager components, as well as collect service and desktop client logs. This connection test is designed to assist you when the location of SQL Diagnostic Manager components changes due to a migration.



You can perform the following tests using system diagnostics:

Management Service Tests


- GUI to management service connection
- GUI and SQL Diagnostic Manager services using the same repository
- Management service repository connection

Collection Service Tests

- Management service to collection service connection
- Connection service configured to use correct management service
- Collection service can connect to management service
- Collection service heartbeat status


One-click System Diagnostics button:

You can run system diagnostics plus collect service and desktop client logs by clicking on **the Run and Gather Logs** button. This feature also saves the status of each collected log in a text file.

 In a SQL Diagnostic Manager distributed environment, the **Run and Gather Logs** command, gathers only local logs. To collect all product logs, run this option from the machine where your SQL Diagnostic Manager services are installed.

Access the System Diagnostics window:

To open the System Diagnostics window, select **Help > System diagnostics** from the Toolbar menu.

 In a console-only installation the **Service Configuration Tool** button is not available since the SQLdm Management Service Configuration Wizard is installed in the machine where all the SQL Diagnostic Manager services reside.

Choose to migrate or recover SQL Diagnostic Manager

When choosing whether a migration or disaster recovery strategy is best for your environment, consider if you want to permanently move the SQL Diagnostic Manager Repository or Services to another computer.

You can also choose to have the Repository and Console components located on computers that belong to different domains. This deployment option is advantageous during disaster recovery, especially if the only available computer is in another domain. Verify that a two-way trust exists between the target domains to support full communication between the Repository and consoles.

Goal	Action
Replace the SQL Diagnostic Manager Repository or Services computer	Migration
Repair the original Repository or Services computer	Recovery

In both cases, first backup and then restore the Repository database.

Create your migration or recovery plan

Once you decide whether a SQL Diagnostic Manager Migration or Recovery is best for your environment, create a plan so that when the time comes, you are prepared and get SQL Diagnostic Manager up and running as quickly as possible.

Creating a Migration Plan

A migration plan details moving the SQL Diagnostic Manager Repository and Services to another SQL Server instance, thereby replacing the original components. You can use a migration plan to respond to an immediate maintenance need. Use the procedures and guidelines in this document to implement or modify your migration plan.

Creating a Recovery Plan

A disaster recovery plan details the steps to remedy unexpected outages to make sure you can continue monitoring SQL Server activity and performance metrics. This document addresses disaster recovery best-practices for establishing a new Repository.

When you implement SQL Diagnostic Manager in your production SQL Server environment, consider preparing a disaster recovery plan to minimize audit data loss should the SQL Diagnostic Manager Repository become unavailable. Use the procedures and guidelines covered in this document to implement or modify your disaster recovery plan.

Understanding the Repository Database

The SQL Diagnostic Manager Repository consists of a SQL Server database named, by default, SQLdmRepository. This database contains the following information:

- Performance metrics and statistics collected from your monitored SQL Server instances
- Configuration settings such as the connection information for the SQL Diagnostic Manager services and alert notification rules
- Alert messages

By default, the Repository database uses the simple recovery model. When this setting is enabled, SQL Server does not maintain the transaction logs for the database. Likewise, any existing transaction logs are not included in backup data. If your corporate policies require transaction log backups, consider changing the recovery model to full so that transaction logs are maintained and archived.

Understanding SQL Diagnostic Manager services

SQL Diagnostic Manager has two centralized services, the Management Service and the Collection Service. These two services reside on the same computer.

The Management Service performs the following primary functions:

- Provides real-time data to the SQL Diagnostic Manager Console
- Receives historical data from the Collection Service for storage in the Repository
- Raises alerts and sends alert notifications

The Collection Service performs on-demand and scheduled collection from the monitored SQL Server instances.

Recovery and Migration Best Practices

Verify the Configuration of the Target SQL Server

When identifying the new SQL Server instance that you want to host the Repository and Services, make sure this instance meets or exceeds the product requirements as well as these specific requirements.

- You can dedicate the target instance to hosting the SQL Diagnostic Manager only
- The target instance runs the same version of SQL Server software that is currently running on the existing SQL Diagnostic Manager computer

Back up the Repository Database

Use a tool such as IDERA SQL Safe to perform a full backup of the Repository database. If you changed the default recovery model to full, make sure your backup includes all transaction logs.

Identify how often to backup the Repository database

The frequency at which you backup the Repository database depends on the following factors:

- How often your alert settings change
- How often your SQL Server environment changes as you add new servers and databases or remove older servers and database
- How much performance data you collect in a given time period
- How much risk you are willing to incur

The backup frequency should reflect your maintenance needs and allow you to meet future monitoring requirements.

Schedule routine backups of the Repository database

After you identify the appropriate backup frequency for your monitoring needs, use a tool such as IDERA SQL Safe to schedule routine backups of the Repository database. If you changed the default recovery model to full, make sure your backup includes all transaction logs.

Review disaster recovery guidelines


Make sure your recovery strategy includes plans to reinstate the original computer once it is repaired.

Consider the following guidelines:

- To minimize data loss, plan to back up the Repository database on the temporary Repository computer immediately before reinstating the original computer.
- Use the procedures in this Technical Solution to reinstate the Repository on the original computer and configure the Management Service.
- To verify all components were reinstated correctly, test your implementation.
- Uninstall the components you previously implemented on the temporary computer.

Migrate your SQL Diagnostic Manager installation

The procedures in this section detail the migration steps for moving both the SQL Diagnostic Manager Repository and the Services to a new computer. Follow each section in order for a successful migration.

 SQL Diagnostic Manager Application security is not maintained when your SQL Diagnostic Manager installation is migrated to a new server. For more information, see [Use application security to manage SQL Diagnostic Manager](#).

Get a New SQL Diagnostic Manager License Key

Because you are migrating the Repository database to a different SQL Server instance, SQL Diagnostic Manager requires a new license key. You can easily request a new license key by sending an email to: licensing@idera.com.

To complete the request, make sure you have the name of the SQL Server instance where you plan to install the SQL Diagnostic Manager Repository database.

Restore the SQL Diagnostic Manager Repository on the Target Computer

For best results, use the following guidelines migrate your SQL Diagnostic Manager Repository to the target computer:

- Perform a full restore
- Schedule the restore during off-hours, or times when you expect the least collection activity

To restore the Repository database:

1. Use SQL Server Enterprise Manager or Management Studio to close any open connections to the Repository database.
2. Use SQL Server Enterprise Manager or Management Studio to take the Repository database offline. If you cannot take the Repository database offline, stop the Collection Service.
3. Use a tool such as IDERA SQL Safe to restore the Repository database on the target server using the appropriate backup archive file.

Disable or Uninstall the SQL Diagnostic Manager Services on the Current Computer

SQL Diagnostic Manager services must be uninstalled to avoid conflict with the restored SQL Diagnostic Manager Repository. For best results, use the following procedure:

If you do not want to keep the SQL Diagnostic Manager console on the current computer, perform the following steps:

1. Select **Start > Control Panel > Uninstall a Program** on your current computer.
2. Uninstall SQL Diagnostic Manager completely.

If you want to keep the SQL Diagnostic Manager Console on the current computer, perform the following steps:

1. Select **Start > Administrative Tools > Services** on the current computer.
2. Scroll down and right-click SQLDM Collection Service in the list, and select **Properties**.
3. In the Startup Type drop-down list, select **Disabled**.
4. Click **OK**.
5. Scroll down and right click SQLDM Management Services in the list, and select **Properties**.
6. In the Startup Type drop-down list, select **Disabled**.

7. Click **OK**.

Perform a Full SQL Diagnostic Manager Install on the New Computer

A full install of SQL Diagnostic Manager installs the new SQL Diagnostic Manager services, link them to the restored SQL Diagnostic Manager Repository, and install the SQL Diagnostic Manager Console. For best results, use the following procedure for installing SQL Diagnostic Manager:

1. Log on with an administrator account to the computer on which you want to install the SQL Diagnostic Manager services.
2. Run Setup.exe in the root of the installation kit.
3. Click **SQL diagnostic manager** on the **Quick Start** window.
4. On the **Welcome** window of the setup program, click **Next**.
5. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click **Next**.
6. Accept the default folder for your SQL Diagnostic Manager installation, or click **Change** to specify a different folder, and then click **Next**.
7. Select your setup type and then click **Next**.
8. Browse and select the SQL Server instance your SQL Diagnostic Manager Repository was restored to and click **Next**.
9. Enter your service account credentials and click **Next**.
10. Click **Install** to begin installing SQL Diagnostic Manager with the options you have selected.

Verify that the Migration was Successful

The SQL Diagnostic Manager Console includes a system diagnostics test that allows you to test the connections of all the SQL Diagnostic Manager components.

1. Open the SQL Diagnostic Manager Console.
2. Enter your new SQL Diagnostic Manager license key.
3. Add the SQL Server instances you want to monitor.
4. Open the System Diagnostics utility by selecting **Help > System Diagnostics**.
5. Click **Test**.
6. Verify that all the tests are successful.

Recovering your SQL Diagnostic Manager installation

In the event that your existing SQL Diagnostic Manager Repository is no longer available, you can recover your SQL Diagnostic Manager installation back to the last point a backup occurred.

Recovering the SQL Diagnostic Manager Repository

To recover lost or damaged data, restore the SQL Diagnostic Manager Repository database using the following guidelines:

- Perform a full restore
- Schedule the restore during off-hours, or times when you expect the least collection activity

To restore the Repository database:

1. Use SQL Server Enterprise Manager or Management Studio to close any open connections to the Repository database.
2. Use SQL Server Enterprise Manager or Management Studio to take the Repository database offline. If you cannot take the Repository database offline, stop the Collection Service.
3. Use a tool such as IDERA SQL Safe to restore the Repository database using the appropriate backup archive file.
4. On the computer hosting the SQL Diagnostic Manager services, restart the SQLDM Collection and Management services.
5. On all existing SQL Diagnostic Manager consoles, reconnect to the SQL Diagnostic Manager Repository by selecting **File >Connect to SQLDM Repository**.

Recovering the SQL Diagnostic Manager services

To install the SQL Diagnostic Manager services:

1. Log on with an administrator account to the computer on which you want to install the SQL Diagnostic Manager services.
2. Run Setup.exe in the root of the installation kit.
3. Click **SQL diagnostic manager** on the **Quick Start** window.
4. On the **Welcome** window of the setup program, click **Next**.
5. Review and accept the license agreement by clicking **I accept the terms in the license agreement**, and then click **Next**.
6. Accept the default folder for your SQL Diagnostic Manager installation, or click **Change** to specify a different folder, and then click **Next**.
7. Select the **Custom Install** setup type, and then click **Next**.
8. Select **SQLDM Services** (also select the **Desktop Client** if you want to run it on the same computer), and then click **Next**.
9. Specify which SQL Server instance is hosting the Repository database, verify that the displayed database name is correct, and then click **Next**.
10. Specify the account credentials that you want the SQL Diagnostic Manager services to use to connect to the SQL Diagnostic Manager Repository and the monitored SQL Server instances, and then click **Next**. This account must belong to the sysadmin role on the SQL Diagnostic Manager Repository database and each instance you plan to monitor.
11. Click **Install** to begin your installation.
12. Click **Finish** to exit the setup program.

Migrate the SQL Diagnostic Manager Repository

The procedures in this section detail the migration steps for moving the SQL Diagnostic Manager Repository to a new computer while leaving the SQL Diagnostic Manager services. Follow each section in order for a successful migration.

Get a New SQL Diagnostic Manager License Key

Because you are migrating the Repository database to a different SQL Server instance, SQL Diagnostic Manager requires a new license key. You can easily request a new license key by sending an email to: licensing@idera.com.

To complete the request, you need the name of the SQL Server instance where you plan to install the SQL Diagnostic Manager Repository database.

Restore the SQL Diagnostic Manager Repository on the Target Computer

To recover lost or damaged data, restore the Repository database. For best results, use the following guidelines:

- Perform a full restore
- Schedule the restore during off-hours, or times when you expect the least collection activity

To restore the Repository database:

1. Use SQL Server Enterprise Manager or Management Studio to close any open connections to the Repository database.
2. Use SQL Server Enterprise Manager or Management Studio to take the Repository database offline. If you cannot take the Repository database offline, stop the Collection Service.
3. Use a tool such as IDERA SQL Safe to restore the Repository database using the appropriate backup archive file.

Configure the Management Service

Configure the Management Service so that it can communicate with the new Repository.

To configure the SQLDM Management Service:

1. Start the Management Service Configuration wizard by selecting **Start > All Programs > IDERA > IDERA SQL Diagnostic Manager > Tools > Management Service Configuration Wizard** on the computer that hosts the SQL Diagnostic Manager services.
2. Read the Welcome window, and then click **Next**.
3. Verify that the following settings are correct, and then click **Next**.
 - The name of the SQL Server instance hosting the Repository database
 - The name of the Repository database
 - The type of authentication the Management Service should use to connect to the Repository database
4. Click **Test** to validate your settings.
5. Click **Finish** to exit the wizard.

Configure the SQLDM Desktop Client

Any previously deployed SQLDM Desktop Client has to be reconfigured to connect to the new Repository.

To reconfigure the SQLDM Desktop Client:

1. Launch the SQLDM Desktop Client by selecting **Start > All Programs > IDERA > IDERA SQL Diagnostic Manager > SQL Diagnostic Manager**.
2. Click **File > Connect to SQLDM Repository**
3. Verify that the following settings are correct, and then click **Next**.
 - The name of the SQL Server instance hosting the Repository database
 - The name of the Repository database
 - The type of authentication the Management Service should use to connect to the Repository database
4. Click **Connect** to save the changes.
5. Upon connecting to the Repository database, SQL Diagnostic Manager will generate a prompt indicating that a valid license does not exist. Click **Manage Licenses**.
6. In the **New Key** section, enter the new license key that was generated for the migration and click **Enter**.
7. Once the key has been added successfully, click the **Close** button.

Verify that the Migration was Successful

The SQL Diagnostic Manager Console includes a system diagnostics test that allows you to test the connections of all the SQL Diagnostic Manager components.

1. Open the SQL Diagnostic Manager console.
2. Enter your new SQL Diagnostic Manager license key.
3. Add the SQL Server instances you want to monitor.
4. Open the System Diagnostics utility by selecting **Help > System Diagnostics**.
5. Click **Test**.
6. Verify that all the tests are successful.

Migrate the SQL Diagnostic Manager services

The procedures in this section detail the migration steps for moving the SQL Diagnostic Manager services to a new computer while leaving the SQL Diagnostic Manager Repository. Follow each section in order for a successful migration.

Uninstall SQL Diagnostic Manager on the Original Computer

Uninstalling SQL Diagnostic Manager on the original computer via Add or Remove Programs, removes all the SQLDM components from the original computer except for the SQL Diagnostic Manager Repository. For best results, use the following procedure for uninstalling SQL Diagnostic Manager on the original computer:

1. Select **Start > Control Panel > Add or Remove Programs**.
2. Scroll down to SQL Diagnostic Manager and select **Remove**.
3. Click **Yes** to the prompt asking if you want to completely remove SQL Diagnostic Manager.

Perform a Full SQL Diagnostic Manager Install on the New Computer

A full install of SQL Diagnostic Manager installs the new SQL Diagnostic Manager services, link them to the restored SQL Diagnostic Manager Repository, and install the SQL Diagnostic Manager Console. For best results, use the following procedure for installing SQL Diagnostic Manager:

1. Log on with an administrator account to the computer on which you want to install the SQL Diagnostic Manager services.
2. Run `Setup.exe` in the root of the installation kit.
3. Click **SQL Diagnostic Manager** on the **Quick Start** window.
4. On the **Welcome** window of the setup program, click **Next**.
5. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click **Next**.
6. Accept the default folder for your SQL Diagnostic Manager installation, or click **Change** to specify a different folder, and then click **Next**.
7. Select your setup type and then click **Next**.
8. Browse and select the SQL Server instance on the original computer that contains your SQL Diagnostic Manager Repository, and click **Next**.
9. Enter your service account credentials and click **Next**.
10. Click **Install** to begin installing SQL Diagnostic Manager with the options you have selected.

Verify that the Migration was successful

The SQL Diagnostic Manager Console includes a system diagnostics test that allows you to test the connections of all the SQL Diagnostic Manager components.

1. Open the SQL Diagnostic Manager Console.
2. Enter your new SQL Diagnostic Manager license key.
3. Add the SQL Server instances you want to monitor.
4. Open the System Diagnostics utility by selecting **Help > System Diagnostics**.
5. Click **Test**.
6. Verify that all the tests are successful.

Deploy SQL Diagnostic Manager in a clustered environment

SQL Diagnostic Manager allows you to monitor your clustered SQL Server environment. See the following topics for installation and configuration instructions for Windows Server 2003, Windows Server 2008, Windows Server 2012, and later environments.

- [Installing in a Windows Server 2003 Clustered Environment](#)
- [Installing in a Windows Server 2008 Clustered Environment](#)
- [Installing in a Windows Server 2012+ Clustered Environment](#)
- [Configure alerts for important cluster events](#)

Windows Server 2003 clustered environment

The following instructions guide you through the installation of SQL Diagnostic Manager in a Windows Server 2003 based clustered environment.

Things to consider before you install SQL Diagnostic Manager in a clustered environment

- Create a separate cluster resource group to host SQLDM services. A separate group allows the SQLDM services to fail without causing other resources in the group to failover
- Before you create this separate cluster resource group, gather a dedicated IP, NetBIOS name, and a dedicated disk for the cluster resource group
- SQLdm Collection, SQLdm Management, SQLdm Predictive Analytics, and SQLdm Rest services can run as generic services on your newly-created cluster resource group

Install SQL Diagnostic Manager on a node

SQL Diagnostic Manager must be installed on each node within the cluster and configure the services as its described in the following page, [Install SQL Diagnostic Manager on a node](#).

Configure the Cluster Resource Group

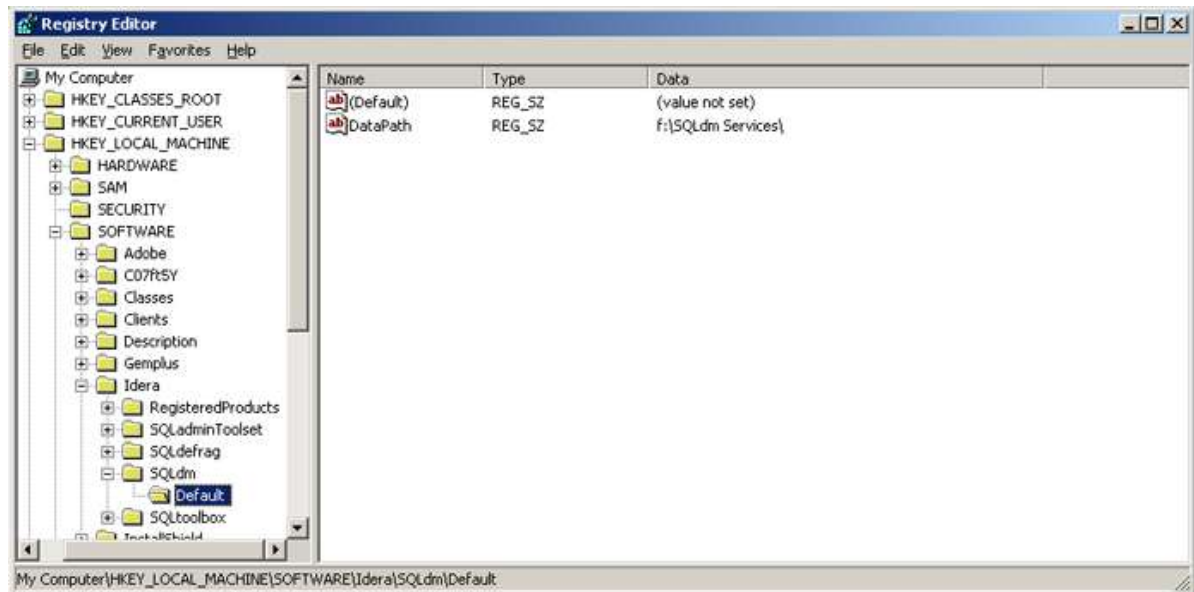
Cluster Resource Group settings must be modified to allow SQL Diagnostic Manager to recognize the cluster nodes.

To configure the Cluster Resource Group:

1. In the Cluster Administrator tool, create the Cluster Resource Group:
 - Select **File > New > Group** and give the group and description
 - Add the nodes and click **Finish**
2. In the Cluster Administrator tool, add a disk to the Cluster Resource Group:
 - Right-click the group you just created, select **New > Resource** and give it a name and description
 - For the Resource Type, select **Physical Disk**
 - Select the SQLdm group
 - Click **Next** until you get to the last window and click **Finish**
3. In the Cluster Administrator tool, add an IP address for the Cluster Resource Group:
 - Right-click the group and select **New > Resource** and give it a name and description
 - Select the Resource type **IP Address**
 - Enter the IP address that you acquired for your new Cluster Resource Group
 - Click **Finish**
4. In the Cluster Administrator tool, add the Network Name resource to the Cluster Resource Group:
 - Right-click the group and **New > Resource** and give it a name
 - Select the Resource type **Network Name**
 - Add the dependency for the IP address resource
 - Specify the name and click **Finish**
5. Select a **Disk** that is in a Cluster Resource Group and create a folder on that disk used to hold local data for the services
6. On the primary node, use Registry Editor to create a new key named **Default** under `HKEY_LOCAL_MACHINE\SOFTWARE\Idera\SQLdm\`
7. Create a new String Value named **DataPath**.
8. Select a **Cluster Resource Group** in which you want to define the SQLdm resources
9. Create an IP Address resource for the SQLDM Services. If there is already an existing Network Name in the Cluster Resource Group, then you can skip this step

- Create an Network Name resource that maps to the IP Address
 - Create the Network Name resource and add the IP Address as a dependency
10. Create a Generic Service resource for the SQLdm Management Service
 - Add the **Disk** for the shared data as a dependency
 - Add the **Network Name** as a dependency
 - Specify SQLdmManagementService\$Default as the name of the service to start
 11. After the service is added, right-click it and select **Properties**
 - Select the Parameters tab and check the **Use network name for computer name** box
 - Select the Registry Replication tab and add *SOFTWARE\Idera\SQLdm\Default*
 - Click **OK**
 12. Create a Generic Service resource for the SQLDM Collection Service
 - Add the **Disk** for the shared data as a dependency
 - Add the **Network Name** as a dependency
 - Specify SQLdmCollectionService\$Default as the name of the service to start
 13. After the service is added, right-click it and select **Properties**
 - Select the Parameters tab and check the **Use network name for computer name** box
 - Select the Registry Replication tab and add *SOFTWARE\Idera\SQLdm\Default*
 - Click **OK**
 14. Create a Generic Service resource for the SQLdm Predictive Analytics Service
 - Add the **Disk** for the shared data as a dependency
 - Add the **Network Name** as a dependency
 - Specify SQLdmPredictiveAnalyticsService\$Default as the name of the service to start
 15. After the service is added, right-click it and select **Properties**
 - Select the Parameters tab and check the **Use network name for computer name** box
 - Select the Registry Replication tab and add *SOFTWARE\Idera\SQLdm\Default*
 - Click **OK**
 16. Create a Generic Service resource for the SQLdm Rest Service
 - Add the **Disk** for the shared data as a dependency
 - Add the **Network Name** as a dependency
 - Specify SQLdmRestService\$Default as the name of the service to start
 17. After the service is added, right-click it and select **Properties**
 - Select the Parameters tab and check the **Use network name for computer name** box
 - Select the Registry Replication tab and add *SOFTWARE\Idera\SQLdm\Default*
 - Click **OK**
 18. In Windows Services, bring the SQLDM Management service resource online
 19. In Windows Services, bring the SQLDM Collection service resource online
 20. In Windows Services, bring the SQLDM Predictive Analytics service resource online

21. In Windows Services, bring the SQLDM Rest service resource online



Complete the Cluster Configuration in SQL Diagnostic Manager

1. Use the Management Service Configuration wizard to force the services to re-register
2. Start the SQLDM Desktop Client and select SQLDM Repository. If the SQLDM Repository is clustered make sure to use the virtual server name for the SQL Server hosting the SQLDM Repository
3. Verify that your configuration displays correctly in the SQLDM Desktop Client

Windows Server 2008 clustered environment

The following instructions guide you through the installation of SQL Diagnostic Manager in a Windows Server 2008 based clustered environment.

Things to consider before you install SQL Diagnostic Manager in a clustered environment

- Create a separate cluster resource group to host SQLDM services. A separate group allows the SQLDM services to fail without causing other resources in the group to failover.
- Before you create this separate cluster resource group, gather a dedicated IP, NetBIOS name, and a dedicated disk for the cluster resource group.
- SQLdm Collection, SQLdm Management, SQLdm Predictive Analytics, and SQLdm Rest services can run as generic services on your newly-created cluster resource group.

Install SQLDM Services on Cluster Nodes

SQL Diagnostic Manager must be installed on each node within the cluster and configure the services as its described in the following page, [Install SQL Diagnostic Manager on a node](#).

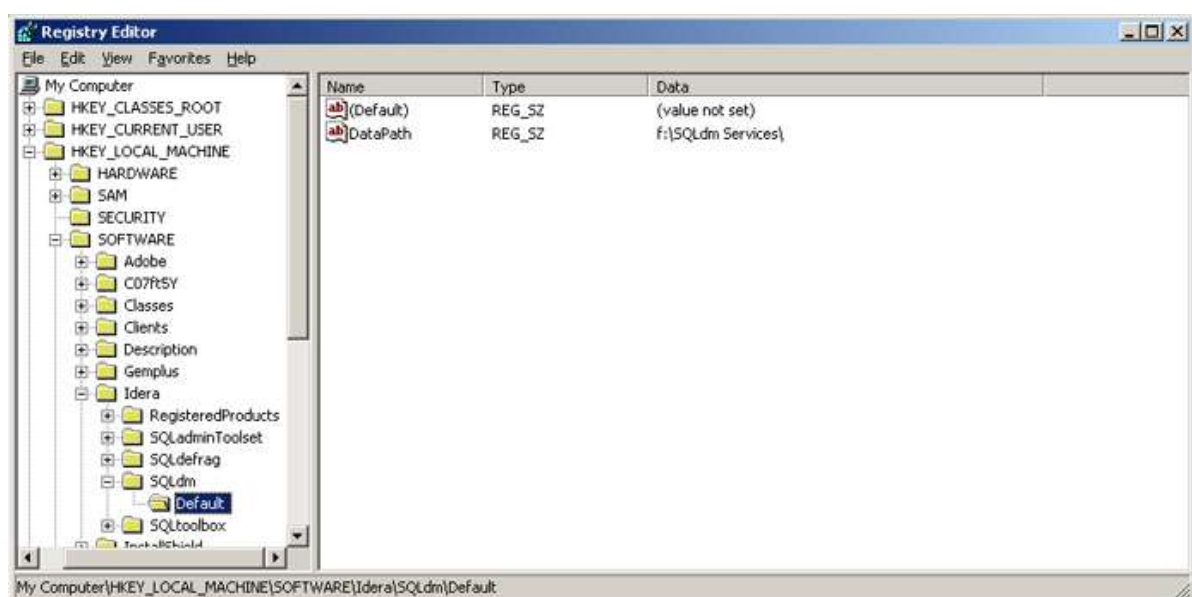
Configure the Cluster Resource Group

Cluster Resource Group settings must be modified to allow SQL Diagnostic Manager to recognize the cluster nodes.

To configure the Cluster Resource Group:

1. Select a disk that is configured in the Cluster Resource Group and create a folder on that disk that you want to use to hold local data for the services
2. On the primary node, edit the registry and add a new key named Default to the key `\SOFTWARE\Idera\SQLdm`. In this key add a string value called **DataPath**. Set its value to the full path to the data directory created in step 1 (as shown in the image at the end of the steps)
3. In the Failover Cluster Management tool, right-click on **Services and Applications** and select **Configure a Service or Application**
 - Provide a name for the service
 - Select **Generic Service**
 - Select the SQLdm Management Service
 - Enter the network name and IP address for the services
 - Select the clustered storage that hosts the path defined in step 1
 - Skip the Registry Replication
 - Verify the settings and click **Next** to complete the wizard
 - Repeat step 3 for the remaining SQLdm services.
4. In **Services and Applications**, click the newly created service name in the Other Resources section and then on the SQLdm Management service and take it offline
5. Select the SQLdm Management Service from the list of resources
6. Select **Next** and **Finish** to exit the wizard
7. In **Services and Applications**, right-click on the newly created service and then on the **SQLdm Management Service** and select Properties
 - On the General tab, select **Use Network Name for computer name**
 - On the Dependencies tab, add the name of service and disk, and then click **Apply**
 - On the Registry Replication tab, if blank, add `SOFTWARE\Idera\SQLdm\Default\`
8. In **Services and Applications**, right-click the newly created service and select **Add a Resource and Generic Service**
9. Select the SQLdm Collection Service from the list of resources

10. Select **Next** and **Finish** to exit the wizard
11. In **Services and Applications**, right-click on the newly created service and then on the **SQLdm Collection Service** and select **Properties**
 - On the General tab, select **Use Network Name for computer name**
 - On the Dependencies tab, add name of service and disk, and then click **Apply**
 - On the Registry Replication tab, add `SOFTWARE\Idera\SQLdm\Default\`
12. In **Services and Applications**, right-click the newly created service and select **Add a Resource and Generic Service**
13. Select the SQLdm Predictive Analytics service from the list of resources
14. Select **Next** and then **Finish** to exit the wizard
15. In **Services and Applications**, click the newly-created service and then on the **SQLdm Predictive Analytics service** and select **Properties**
 - On the General tab, select **Use Network Name for computer name**
 - On the Dependencies tab, add the name of the service and disk, and then click **Apply**
 - On the Registry Replication tab, if blank, add `SOFTWARE\Idera\SQLdm\Default\`
16. In **Services and Applications**, right-click the newly created service and select **Add a Resource and Generic Service** .
17. Select the SQLdm Rest service from the list of resources
18. Select **Next** and then **Finish** to exit the wizard
19. In **Services and Applications**, click the newly-created service and then on the **SQLdm Rest service** and select **Properties**
 - On the General tab, select **Use Network Name** for computer name
 - On the Dependencies tab, add the name of the service and disk, and then click **Apply**
 - On the Registry Replication tab, if blank, add `SOFTWARE\Idera\SQLdm\Default\`
20. In Windows Services, bring the SQLdm Management service resource online
21. In Windows Services, bring the SQLdm Collection service resource online
22. In Windows Services, bring the SQLdm Predictive Analytics service resource online
23. In Windows Services, bring the SQLdm Rest service resource online
24. On the active node on which you installed the SQLdm services, use the Management Service Configuration wizard (**Start > All Programs > IDERA > SQL Diagnostic Manager > Tools > Management Service Configuration Wizard**) to force the services to re-register
25. Test the configuration from the Management Service Configuration wizard before finishing the wizard



Complete the Cluster Configuration in SQL Diagnostic Manager

Start the SQLDM Desktop Client and select SQLDM Repository. If the SQLDM Repository is clustered make sure to use the virtual server name for the SQL Server hosting the SQLDM Repository.

Windows Server 2012+ clustered environment

The following instructions guide you through the installation of SQL Diagnostic Manager in a Windows Server 2012 and later based clustered environment.

Things to consider before you install SQL Diagnostic Manager in a clustered environment

- Create a separate cluster resource group to host SQLDM services. A separate group allows the SQLDM services to fail without causing other resources in the group to failover
- To create a cluster resource group you need to gather a dedicated IP, NetBIOS name, and a dedicated disk
- SQLdm Collection, SQLdm Management, SQLdm Rest Service, and Predictive Analytics services can run as generic services on your newly-created cluster resource group

Next, follow the steps described in the links below to complete the installation of SQL Diagnostic Manager on a Windows Server 2012+ clustered environment:

1. [Install SQL Diagnostic Manager on a node](#)
2. [Register the SQL Diagnostic Manager services as a Clustered Resource](#)
3. [Reconfigure the SQL Diagnostic Manager Services and Repository Database](#)
4. [Install the IDERA Dashboard and Register the SQL Diagnostic Manager with the IDERA Dashboard](#)

Install SQL Diagnostic Manager on a node

SQL Diagnostic Manager must be installed on each node within the cluster. Perform the following steps on each node:

1. On a node of the cluster, download the SQL Diagnostic Manager installation kit ZIP file
2. Extract and run the executable file. Once, the extraction of the files completes, the IDERA SQL Diagnostic Manager Quick Start window displays
3. Click the **Install SQL Diagnostic Manager** link
4. On the Welcome to the SQL Diagnostic Manager Setup, click **Next**
5. Review the license agreement. To accept this agreement, check the box I accept the terms and conditions of this License Agreement, and click **Next**
6. On the Customize your SQLdm setup screen, select both SQLDM Console and SQLDM Services and Repository options, and click **Next**
7. On the Registration with IDERA Dashboard screen, select the **Proceed without registration** option, and click **Next**
8. On the Service Account, enter your credentials that are used by the SQLDM services, and click **Next**
9. Select the destination directory on the Choose a destination folder screen. Click **Next**
10. Click **Install**, to start the installation of SQL Diagnostic Manager on the node
11. When the installation is completed, uncheck the Launch Idera SQL Diagnostic Manager selection, and click **Finish**
12. Open Windows Services
13. Locate and stop the following services:
 - a. **SQLdm Collection Service**
 - b. **SQLdm Management Service**
 - c. **SQLdm Predictive Analytics Service**
 - d. **SQLdm Rest Service**
14. Update the properties for all the mentioned services above, to Manual startup type
15. Repeat the steps above for the remaining nodes of the cluster

Once you completed the SQL Diagnostic Manager installation for all the cluster nodes, continue with your the configuration of the Cluster Resource group according to your Windows Server version:

- Windows Server 2003 clustered environment: [Configure the Cluster Resource Group](#)
- Windows Server 2008 clustered environment: [Configure the Cluster Resource Group](#)
- Windows Server 2012+ clustered environment: [Register the SQL Diagnostic Manager Services as a Clustered Resource](#).

Register SQL Diagnostic Manager services as a Clustered Resource


Registering SQL Diagnostic Manager services with Microsoft Failover Cluster Manager allows the Microsoft Cluster Service to manage the services in failover situations. The following configuration ensures the high availability of the services during a failover.

Below, you can find two sets of instructions to register the SQL Diagnostic Manager services as a clustered resource:

- [Creating a new role for SQL Diagnostic Manager services](#)
- [Adding the SQL Diagnostic Manager services to an existing role](#)

Creating a new role for SQL Diagnostic Manager services

This set of instructions helps you to configure SQL Diagnostic Manager services in its own cluster. This option requires an available disk on the cluster to store the temporary files used by the SQLDM services.

 An available disk on the cluster can be any visible disk to all nodes in the cluster; including a network disk reachable via the UNC path.

1. Launch **Failover Cluster Manager** on a node of the cluster. For more information on how to launch the Failover Cluster Manager, check [Installing the Failover Cluster Feature and Tools in Windows Server 2012](#)
2. Right-click on **Role** and select **Create Empty Role**
3. Right-click on **New Role** and select **Properties**
4. Specify a new name for the role and click **OK**
5. Right-click on the **Role**, select **Add Resource**, and click **Client Access Point**
6. When the New Resource Wizard opens, type a unique network name and an IP address, and click **Next**
7. Click **Next** on the Confirmation screen
8. Click **Finish** on the Summary screen
9. Right-click on the Role and select **Add Storage**
10. Select a disk, and click **OK**
11. In File Explorer, create a folder to be used by SQLDM services on the disk that was added
12. Open Registry Editor and navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Idera\SQLdm`
13. Right-click the SQLdm key, select **New**, click **Key**, and enter **Default** for the name
14. Right-click the Default key, select **new**, click **String Value**, and enter **DataPath** for the name
15. Double-click the string value and enter the path to the folder that was created for SQLDM services, click **OK**
16. Go back to Failover Cluster Manager
17. Right-click on the **role**, select **Add Resource**, and click **Generic Service**
18. On the Select Service screen, select the **SQLdm Collection Service**, and click **Next**
19. Click **Next** on the Confirmation screen
20. Click **Finish** on the Summary screen
21. Repeat steps 17-20 for the remaining SQLDM services: SQLdm Management Service, SQLdm Predictive Analytics Service, and SQLdm Rest Service
22. Right-click on one of the added SQLDM services and select **Properties**
23. Go to **Dependencies** tab and add the following resources:
 - a. Name
 - b. IP Address
 - c. Disk
24. Click **Apply** and go to the **General** tab
25. Check the box of Use network name for computer name, and click **Apply**
26. Go to the Registry Replication tab, click **Add**, enter `SOFTWARE\Idera\SQLdm\Default`, and click **OK**

⚠ There is a bug in Window 2012 where the Registry Replication tab is not available. If the tab is unavailable, use the [Add-ClusterCheckpoint](#) PowerShell cmdlet to add the necessary setting.
 EXAMPLE: Add-ClusterCheckpoint -ResourceName "SQLDmCollectionService`\$Default"
 -RegistryCheckpoint "SOFTWARE\Idera\SQLDm\Default" -Cluster "SQLDmServices"

27. Click **Apply**, and then click **OK**
28. Repeat steps 17-27 for the remaining SQLDM services: SQLdm Management Service, SQLdm Predictive Analytics Service, and SQLdm Rest Service
29. Right-click on each SQLDM service and select **Bring Online**

Adding the SQL Diagnostic Manager services to an existing role

The following set of instructions helps you to add the SQL Diagnostic Manager services to an existing role:

1. In File Explorer, create a folder to be used by SQLDM services on a disk that has been added as a cluster resource.
2. Open Registry Editor and navigate to *HKEY_LOCAL_MACHINE\SOFTWARE\Idera\SQLdm*
3. Right-click the SQLdm key, select **New**, click **Key**, and enter **Default** for the name
4. Right-click the Defaultkey, select **new**, click **String Value**, and enter **DataPath** for the name
5. Double-click the string value and enter the path to the folder that was created for SQLDM services, click **OK**
6. Launch **Failover Cluster Manager** on a node of the cluster. For more information on how to launch the Failover Cluster Manager, check [Installing the Failover Cluster Feature and Tools in Windows Server 2012](#)
7. Right-click on an existing role, select **Add Resource**, and click **Generic Service**
8. On the Select Service screen, select **SQLDmCollection Service**, and click **Next**
9. Click Next on the Confirmation screen
10. Click Finish on the Summary screen
11. Repeat steps 7-10 for the remaining SQLDM services: SQLdm Management Service, SQLdm Predictive Analytics Service, and SQLdm Rest Service
12. Right-click on one of the added SQLDM service, and select **Properties**
13. Go to **Dependencies** tab and add the following resources:
 - a. Name
 - b. IP Address
 - c. Disk
14. Click **Apply** and go to the **General** tab
15. Check the box of Use network name for computer name, and click **Apply**.
16. Go to the Registry Replication tab, click **Add**, enter *SOFTWARE\Idera\SQLDm\Default*, and click **OK**

⚠ There is a bug in Window 2012 where the Registry Replication tab is not available. If the tab is unavailable, use the [Add-ClusterCheckpoint](#) PowerShell cmdlet to add the necessary setting.
 EXAMPLE: Add-ClusterCheckpoint -ResourceName "SQLDmCollectionService`\$Default"
 -RegistryCheckpoint "SOFTWARE\Idera\SQLDm\Default" -Cluster "SQLDmServices"

17. Click **Apply**, and then click **OK**
18. Repeat steps 12-17 for the remaining SQLDM services: SQLdm Management Service, SQLdm Predictive Analytics Service, and SQLdm Rest Service
19. Right-click on each SQLDM service and select **Bring Online**

When you finish registering the SQL Diagnostic Manager services as Clustered Resources, you need to [Reconfigure the SQLDM Services and Repository Database](#).

Register SQL Diagnostic Manager with the IDERA Dashboard

IDERA Dashboard does not currently support clustered environments. Given this limitation, IDERA Dashboard must be installed on a stand-alone server. The following details how to install and register SQL Diagnostic Manager in the IDERA Dashboard, allowing users to access the product using a web browser:

Install the IDERA Dashboard on a stand-alone server

1. Log onto a stand-alone server using an administrator account
2. Download the SQL Diagnostic Manager installation kit ZIP file
3. Extract and run the executable file. Once the extraction has completed, click the **Install the SQL Diagnostic Manager** link located in the IDERA SQL Diagnostic Manager Quick Start window
4. When the Welcome to the SQL Diagnostic Manager Setup wizard screen displays, click **Next**
5. Review the License Agreement. To accept this license agreement, click **I accept the terms and conditions of this License Agreement**, and then click **Next**
6. On the Customize your SQLdm setup page, only select the IDERA Dashboard option, and click **Next**
7. Enter the credentials that you need to run the IDERA Dashboard services, and click **Next**
8. When the Service Ports screen displays, accept the defaults ports by clicking **Next**
9. Specify the SQL Server instance to host the repository database for the IDERA Dashboard, and click **Next**
10. Click **Install**, to start the installation of the IDERA Dashboard
11. Once the installation is finished, click the **Finish** button
12. Close the IDERA SQL Diagnostic Manager Quick Start window

Register SQL Diagnostic Manager with IDERA Dashboard

1. Log into the IDERA Dashboard using a Dashboard administrator account

✓ The URL used to access the IDERA Dashboard is constructed from the name of the machine on which the IDERA Dashboard was installed and the port specified for the Idera Dashboard Web Application Service.
For example, if the IDERA Dashboard was installed on a machine named MYSERVER and the Idera Dashboard Web Application Service used the default port of 9290, then the URL to access the IDERA Dashboard would be the following: *http://MYSERVER:9290*

2. Go to Administration, and select **Manage Products**
3. Click the **Register a Product** link
4. When the Add a Product to IDERA Dashboard window displays, fill out the required fields to register SQL Diagnostic Manager:
 - a. Product: SQLDM
 - b. Display name: A unique name of your choice
 - c. Host: The network name that is configured as a dependency for the SQLDM services (e.g. SQLdmServices)
 - d. Port: 5170
 - e. Username and Password: Enter the credentials of a Dashboard administrator account.
5. Click **Register**
6. Click **Yes** to confirm the registration of the product
7. Log into the IDERA Dashboard again to confirm the product has been registered

Deploy IDERA Newsfeed in a clustered environment

You can deploy the SQLDM Mobile & Newsfeed Service to a clustered environment for continuous remote alerting. See the following topics for installation and configuration instructions for Windows Server 2003, Windows Server 2008, and Windows Server 2012 environments.

- [Installing in a Windows Server 2003 Clustered Environment](#)
- [Installing in a Windows Server 2008 Clustered Environment](#)
- [Installing in a Windows Server 2012 Clustered Environment](#)

Deploy the IDERA Newsfeed to a Windows 2003 cluster

Use the following instructions to deploy the IDERA Newsfeed Platform in a Windows Server 2003 clustered environment.

Things to consider before you install the IDERA Newsfeed Platform in a clustered environment

- Assign the IDERA Newsfeed Platform to a new or existing cluster resource group. You may want to create a separate cluster resource group to host the SQLDM Mobile & Newsfeed Service.
 - Creating a new cluster resource group for this deployment allows the service to fail without causing other resources in the cluster to fail over. However, this configuration also requires a new dedicated IP, a new NetBIOS name, and a new dedicated drive.
 - Adding the SQLDM Mobile & Newsfeed Service to your existing SQLDM Services resource cluster group makes sure that all four services fail over together, and does not require additional resources.
- The SQLDM Mobile & Newsfeed Service runs as a generic service on the assigned cluster resource group.
- You may need to provide a Network Name and Network IP during the cluster configuration.
- When you install the IDERA News Platform, you need to provide credentials for the service account and the name of a SQL Server instance that can host the SQLDM Mobile Repository database. Make sure these components meet the [requirements](#).
- Install or upgrade your clustered SQLDM Services to the latest version of SQL Diagnostic Manager.

Install the IDERA Newsfeed Platform on each cluster node

Install the platform on each cluster node to set up the IDERA Newsfeed Platform to work correctly when the primary cluster node fails.

To install IDERA Newsfeed Platform:

1. Log on using an administrator account.
2. Run Setup . exe in the root of the SQLDM installation kit.
3. Click **SQLDM Mobile and Newsfeed** on the **Quick Start** window.
4. On the **Welcome** window of the setup program, click **Next**.
5. Review and accept the license agreement by clicking **I accept the terms in the license agreement**, and then click **Next**.
6. Select the **Mobile & Newsfeed components**, and then click **Next**.
7. Specify the appropriate credentials for the service account, and then click **Next**.
8. Select which SQL Server instance should host the Repository database, and then click **Next**.
9. Specify the account that you want the setup program to use to create the Repository database, test the connection, and then click **Next**.
10. Indicate which credentials you want the SQLDM Mobile & Newsfeed Service account to use to connect to the Repository database, and then click **Next**.
11. Specify which computer hosts your SQLDM Repository, and then click **Next**.
12. Click **Install**.
13. In Windows Services, stop the following services, and set their **Startup type** to Manual.
 - SQLDM Mobile & Newsfeed Service
 - SQLDM Management Service
 - SQLDM Collector Service
 - SQLDM Predictive Analytics

Create the cluster resource group (optional)

For more granular failover control, you can create a dedicated cluster resource group for the IDERA Newsfeed Platform.

To create a cluster resource group for the IDERA Newsfeed Platform:

1. In the Cluster Administrator tool, create the cluster resource group:
 - a. Select **File > New > Group**, and then specify a name and description.
 - b. Add the nodes that belong to this cluster, and then click **Finish**.
2. Add a disk to the cluster resource group:
 - a. Right-click the group you just created, select **New > Resource**, and then specify a name and description.
 - b. For the Resource Type, select **Physical Disk**.
 - c. Select the IDERA Newsfeed Platform group.
 - d. Click **Next** until you get to the last window and click **Finish**.
3. Add an IP address for the cluster resource group:
 - a. Right-click the group, select **New > Resource**, and then specify a name and description.
 - b. Select the Resource type **IP Address**.
 - c. Type the IP address that you acquired for your new cluster resource group.
 - d. Click **Finish**.
4. Add the Network Name resource to the cluster resource group:
 - a. Right-click the group, select **New > Resource**, and then specify a name and description.
 - b. Select the resource type **Network Name**.
 - c. Add the IP address as a dependency for this release.
 - d. Specify a name, and then click **Finish**.

Add the SQLDM Mobile & Newsfeed Service to the appropriate cluster resource group

On each cluster node, modify the cluster resource group settings to allow the IDERA Newsfeed to recognize the cluster nodes.

To add the SQLDM Mobile & Newsfeed Service to your resource cluster group:

1. In the Cluster Administrator tool, navigate to the cluster resource group to which you want to add the IDERA Newsfeed Service as a resource. You can use the new cluster resource group you previously created, or the cluster resource group you configured for your SQLDM Services.
2. Select the disk you created for this cluster resource group, and then create a directory on that disk. The SQLDM Mobile & Newsfeed Service uses this folder to store data.
3. Create an IP Address resource for the SQLDM Mobile & Newsfeed Service. **If a Network Name is already specified for this cluster resource group**, skip this step.
 - Create a Network Name resource (it maps to the IP Address)
 - Add the IP Address as a dependency
4. Create a Generic Service resource for the SQLDM Mobile & Newsfeed Service.
 - Add the **Disk** for the shared data as a dependency
 - Add the **Network Name** as a dependency
 - Specify SQLdmMobileandNewsService\$Default as the name of the service to start
5. After the service is successfully added, update its properties.
 - a. On the Parameters tab, check the **Use network name for computer name** option.
 - b. On the Registry Replication tab, add the SOFTWARE\IDERA\IDERA Newsfeed & Mobile\DataPath key.
 - c. Click **OK**.
6. In Windows Services, bring the following services online:

- SQLDM Mobile & Newsfeed Service
- SQLDM Management Service
- SQLDM Collector Service
- SQLDM Predictive Analytics

Test your cluster configuration using the SQLDM Console

Make sure the SQLDM services can communicate with each other and the Repository databases.

1. On the active node of the cluster, use the Management Service Configuration wizard to re-register the services and test your configuration. You can start this tool from the **Programs** menu.
2. Verify that you are able to create a Newsfeed account and monitor your virtual SQL Server instances using the News Feed interface in the SQLDM Console.

Deploy the IDERA Newsfeed to a Windows 2008 cluster

Use the following instructions to deploy the IDERA Newsfeed Platform in a Windows Server 2008 clustered environment.

Things to consider before you install the IDERA Newsfeed Platform in a clustered environment

- Assign the IDERA Newsfeed Platform to a new or existing clustered service. You may want to create a separate clustered service to host the SQLDM Mobile & Newsfeed Service.
 - Creating a new clustered service for this deployment allows the service to fail without causing other resources in the cluster to failover. However, this configuration also requires a new dedicated IP, a new NetBIOS name, and a new dedicated drive.
 - Adding the SQLDM Mobile & Newsfeed Service to your existing SQLDM Services clustered service makes sure that all four services failover together, and does not require additional resources.
- The SQLDM Mobile & Newsfeed Service runs as a generic service on the assigned clustered service.
- You may need to provide a Network Name and Network IP during the cluster configuration.
- When you install the IDERA News Platform, you need to provide credentials for the service account and the name of a SQL Server instance that can host the SQLDM Mobile Repository database. Make sure these components meet the [requirements](#).
- Install or upgrade your clustered SQLDM Services to the latest version of SQL Diagnostic Manager.

Install the IDERA Newsfeed Platform on each cluster node

To set up IDERA Newsfeed Platform to work correctly when the primary cluster node fails, you need to install the platform on each cluster node.

To install IDERA Newsfeed Platform:

1. Log on using an administrator account.
2. Run Setup . exe in the root of the SQLDM installation kit.
3. Click **SQLDM Mobile and Newsfeed** on the **Quick Start** window.
4. On the **Welcome** window of the setup program, click **Next**.
5. Review and accept the license agreement by clicking **I accept the terms in the license agreement**, and then click **Next**.
6. Select the **Mobile & Newsfeed components**, and then click **Next**.
7. Specify the appropriate credentials for the service account, and then click **Next**.
8. Select which SQL Server instance should host the Repository database, and then click **Next**.
9. Specify the account that the setup program should use to create the Repository database, test the connection, and then click **Next**.
10. Indicate which credentials the SQLDM Mobile & Newsfeed Service account should use to connect to the Repository database, and then click **Next**.
11. Specify which computer hosts your SQLDM Repository, and then click **Next**.
12. Click **Install**.
13. In Windows Services, stop the following services, and set their **Startup type** to Manual.
 - SQLDM Mobile & Newsfeed Service
 - SQLDM Management Service
 - SQLDM Collection Service
 - SQLDM Predictive Analytics

Create the clustered service (optional)

For more granular failover control, you can create a dedicated clustered service for the IDERA Newsfeed Platform.

To configure the clustered service:

1. Start the Registry Editor and navigate to the ...\\IDERA\\SQLdm Mobile & Newsfeed\\ key. For this key, add a **String Value** named DataPath and set it to directory path for the new folder.
2. In the Failover Cluster Management tool, right-click the **Services and Applications** node, and then click **Configure a Service or Application**.
 - a. Provide a name for the new clustered service.
 - b. Select **Generic Service**.
 - c. Select the SQLDM Mobile & Newsfeed Service.
 - d. Enter the network name and IP address for the services.
 - e. Select the clustered storage that you want to host the data folder path.
 - f. Skip the Registry Replication.
 - g. Verify your selections, and then click **Next** to complete the wizard.
3. Under **Other Resources** in the Services and Applications view, right-click on your new clustered service, and then take the SQLDM Mobile & Newsfeed Service offline.

Add the SQLDM Mobile & Newsfeed Service to the appropriate clustered service

On each cluster node, modify the clustered service settings to allow the IDERA Newsfeed Platform to recognize the cluster nodes.

1. Select a disk that is configured for the target clustered service, and create a folder on that disk to store data for the SQLDM Mobile & Newsfeed Service. You can choose your new clustered service or the existing clustered service already used for the SQLDM Services.
2. Expand **Services and Applications**, right-click the newly created service, and then click **SQLDM Mobile & Newsfeed Service > Properties**.
 - On the **General** tab, select **Use Network Name** for the computer name.
 - On the **Dependencies** tab, specify the **Disk** and **Name** used by the service.
 - On the **Registry Replication** tab, add HKLM\\SOFTWARE\\IDERA\\IDERA Newsfeed & Mobile\\DataPath.
3. Right-click the new clustered service node, then select **Add a Resource > Generic Service**.
4. Select the **SQLDM Mobile & Newsfeed Service** from the list of services.
5. Click **Next** to finish the wizard.
6. In Windows Services, bring the following services online:
 - SQLDM Mobile & Newsfeed Service
 - SQLDM Management Service
 - SQLDM Collection Service
 - SQLDM Predictive Analytics

Test your cluster configuration using the SQLDM Console

Make sure the SQL Diagnostic Manager services can communicate with each other and the Repository databases.

1. On the active node of the cluster, use the Management Service Configuration wizard to re-register the services and test your configuration. You can start this tool from the **Programs** menu.
2. Verify that you are able to create a Newsfeed account and monitor your virtual SQL Server instances using the News Feed interface in the SQLDM Console.

Deploy the IDERA Newsfeed to a Windows 2012 cluster

Use the following instructions to deploy the IDERA Newsfeed Platform in a Windows Server 2012 clustered environment.

Things to consider before you install the IDERA Newsfeed Platform in a clustered environment

- Assign the IDERA Newsfeed Platform to a new or existing clustered service. You may want to create a separate clustered service to host the SQLDM Mobile & Newsfeed Service.
 - Creating a new clustered service for this deployment allows the service to fail without causing other resources in the cluster to failover. However, this configuration also requires a new dedicated IP, a new NetBIOS name, and a new dedicated drive.
 - Adding the SQLDM Mobile & Newsfeed Service to your existing SQLDM Services clustered service makes sure that all four services failover together, and does not require additional resources.
- The SQLDM Mobile & Newsfeed Service runs as a generic service on the assigned clustered service.
- You may need to provide a Network Name and Network IP during the cluster configuration.
- When you install the IDERA News Platform, you need to provide credentials for the service account and the name of a SQL Server instance that can host the SQLDM Mobile Repository database. Make sure these components meet the [requirements](#).
- Install or upgrade your clustered SQLDM Services to the latest version of SQL Diagnostic Manager.

Install the IDERA Newsfeed Platform on each cluster node

To set up IDERA Newsfeed Platform to work correctly when the primary cluster node fails, you need to install the platform on each cluster node.

To install IDERA Newsfeed Platform:

1. Log on using an administrator account.
2. Run Setup .exe in the root of the SQLDM installation kit.
3. Click **SQLDM Mobile and Newsfeed** on the **Quick Start** window.
4. On the **Welcome** window of the setup program, click **Next**.
5. Review and accept the license agreement by clicking **I accept the terms in the license agreement**, and then click **Next**.
6. Select the **Mobile & Newsfeed components**, and then click **Next**.
7. Specify the appropriate credentials for the service account, and then click **Next**.
8. Select which SQL Server instance should host the Repository database, and then click **Next**.
9. Specify the account that the setup program should use to create the Repository database, test the connection, and then click **Next**.
10. Indicate which credentials the SQLDM Mobile & Newsfeed Service account should use to connect to the Repository database, and then click **Next**.
11. Specify which computer hosts your SQLDM Services, and then click **Next**.
12. Click **Install**.
13. In Windows Services, stop the following services, and set their **Startup type** to Manual.
 - SQLDM Mobile & Newsfeed Service

Create the clustered service (optional)

For more granular failover control, you can create a dedicated clustered service for the IDERA Newsfeed Platform.

To configure the clustered service:

1. Start the Registry Editor and navigate to the `... \IDERA\SQLdm Mobile & Newsfeed\` key. For this key, add a **String Value** named `DataPath` and set it to directory path for the new folder.
2. In the Failover Cluster Management tool, right-click **Roles**, and then click **Configure Role**.
 - a. Provide a name for the new clustered service.
 - b. Select **Generic Service**.
 - c. Select the SQLDM Mobile & Newsfeed Service.
 - d. Enter the network name and IP address for the services.
 - e. Select the clustered storage that you want to host the data folder path.
 - f. Skip the Registry Replication.
 - g. Verify your selections, and then click **Next** to complete the wizard.
3. Under **Other Resources** in the Services and Applications view, right-click on your new clustered service, and then take the SQLdm Mobile & Newsfeed Service offline.
4. To configure the Registry Replication entry, open PowerShell with administrator rights and execute the following command:


```
Add-ClusterCheckpoint -ResourceName "(Newsfeed Role Name)" -RegistryCheckpoint "SOFTWARE\IDERA\SQLdm Mobile and Newsfeed\Default\DataPath"
```

Add the SQLDM Mobile & Newsfeed Service to the appropriate clustered service

On each cluster node, modify the clustered service settings to allow the IDERA Newsfeed Platform to recognize the cluster nodes.

1. Select a disk that is configured for the target clustered service, and create a folder on that disk to store data for the SQLDM Mobile & Newsfeed Service. You can choose your new clustered service or the existing clustered service already used for the SQLDM Services.
2. Expand **Services and Applications**, right-click the newly created service, and then click **SQLDM Mobile & Newsfeed Service > Properties**.
 - On the **General** tab, select **Use Network Name** for the computer name.
 - On the **Dependencies** tab, specify the **Disk** and **Name** used by the service.
 - On the **Registry Replication** tab, add `HKLM\SOFTWARE\IDERA\IDERA Newsfeed & Mobile\DataPath`.
3. Right-click the new clustered service node, then select **Add a Resource > Generic Service**.
4. Select the **SQLDM Mobile & Newsfeed Service** from the list of services.
5. Click **Next** to finish the wizard.
6. In Windows Services, bring the following services online:
 - SQLDM Mobile & Newsfeed Service
 - SQLDM Management Service
 - SQLDM Collection Service
 - SQLDM Predictive Analytics

Test your cluster configuration using the SQLDM Console

Make sure the SQLDM services can communicate with each other and the Repository databases.

1. On the active node of the cluster, use the Management Service Configuration wizard to re-register the services and test your configuration. You can start this tool from the **Programs** menu.
2. Verify that you are able to create a Newsfeed account and monitor your virtual SQL Server instances using the News Feed interface in the SQLDM Console.

Default alert IDs

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
0	101	Resources	SQL Server CPU Usage (Percent)	Percentage of CPU Time used by the SQL Server instance out of the total on the SQL Server computer.
1	202	Sessions	User Connections (Percent)	Percentage of the user connections allowed by the currently used SQL Server instances.
4	708	Services	Non-distributed Transactions (Count)	Number of replication transactions published by the Publisher but not received by the Distributor.
5	809	Services	Unsubscribed Transactions (Count)	Number of replication transactions received by the Distributor but not received by a Subscriber.
6	910	Queries	Oldest Open Transaction (Minutes)	Time in minutes that the longest running transaction in the database used since issuing the BEGIN TRANSACTION statement.
7	1011	Databases	Table Fragmentation (Percent)	Percentage of pages in the table that are fragmented.
8	1112	Databases	Database Full (Percent)	Percentage of the allowable disk space for the database currently used by the sum of data, text, and indexes. Allowable disk space for the database is calculated by taking into account the current allocated space, auto-growth settings, and available disk space.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
9	1213	Databases	Log Full (Percent)	Percentage of the allowable disk space for the database currently used by the transaction log. Allowable disk space for the database is calculated by taking into account the current allocated space, auto-growth settings and available disk space.
10	1314 - Running 1315 - Stopped 1316 - Other	Services	SQL Server Agent Status	Status of the SQL Server Agent service.
11	1417	Databases	Database Read/Write Error Occurred	An I/O error occurred while attempting to read from or write to a database file.
12	1518 - Running 1519 - Stopped 1520 - Paused 1521 - Not Admin 1522 Not Accessible 1523 - Other	Services	SQL Server Status	Status of the SQL Server service.
13	1624	Resources	SQL Server Memory Usage (Percent)	Percentage of memory used by the SQL Server instance out of the total memory on the SQL Server computer.
14	1725	Databases	Database Status	Operational status of the database, such as Normal, Offline, or Suspect.
17	2028	Services	Unsubscribed Transactions (Seconds)	Time in seconds that a replication transaction is holding at the Distributor and waiting for a Subscriber to receive it. This alert is for visual indicators in the Desktop Client only. No notifications are sent.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
18	2129 - Running 2130 - Stopped 2131 - Other	Services	DTC Status	Status of the Distributed Transaction Coordinator service.
19	2232 - Running 2233 - Other	Services	Full-Text Search Status	Status of the Full-Text Search service.
20	N/A	Databases	Index Row hits (Percent)	Percentage of density of an index as used by the SQL Query Optimizer to determine index effectiveness. A high density indicates many duplicates and an inefficient index while a low density indicates high selectivity and an effective index. This alert is for visual indicators in the Desktop Client only. No notifications are sent.
21	N/A	Services	Last Full-Text Catalog Update (Hours)	Time (in hours) since the last Full-Text Catalog update was performed and beyond which the statistics data is considered outdated. This alert is for visual indicators in the Desktop Client only. No notifications are sent.
22	2534	Sessions	SQL Server Response Time (Milliseconds)	Time (in milliseconds) SQL diagnostic manager currently needs to send a simple SQL command to the SQL Server instance, have it processed, and receive the returned result set.
23	4010 - Available 4011 - Other 4012 - Disabled 4013 - Unavailable	Operational	OS Metrics Collection Status	Status of SQL diagnostic manager OS Metrics Collection.
24	4020	Resources	OS Used Memory (Percent)	Percentage of total memory used on the SQL Server computer.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
25	4030	Resources	OS Paging (Per Second)	Page fault rate for all processes on the SQL Server computer.
26	4040	Resources	OS Processor Time (Percent)	Percentage of CPU time used by all processes on the SQL Server computer.
27	4050	Resources	OS Privileged Time (Percent)	Percentage of CPU time spent by all processes executing in privileged mode on the SQL Server computer.
28	4060	Resources	OS User Time (Percent)	Percentage of CPU time spent by all processes executing in user mode on the SQL Server computer.
29	4070	Resources	OS Processor Queue Length (Count)	Number of ready threads in the processor queue on the SQL Server Computer.
30	4080	Resources	OS Disk Time (Percent)	Percentage of elapsed time that all the disks were busy servicing read and write requests on the SQL Server computer.
31	4090	Resources	OS Average Disk Queue Length (Count)	Average number of both read and write requests queued for all the disks on the SQL Server computer.
32	2737	Sessions	Session CPU Time (Seconds)	CPU Time in seconds used by the SQL Server session since the last scheduled refresh.
33	2838	Sessions	Session CPU Time (Seconds)	Time in seconds that the SQL Server session is blocking other sessions.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
34	2940	Services	SQL Server Agent Long Running Job	Percentage of time a scheduled SQL Server Agent job is taking over the average successful completion time for this job in the current job history.
35	3039	Services	SQL Server Agent Job Failure	The scheduled SQL Server Agent job ended with a Failed status.
48	4001 - Enabled 4002 - Disabled	Operational	Maintenance Mode Enabled	Maintenance Mode is enabled for the SQL Server instance. SQL diagnostic manager does not collect scheduled data or process alerts.
49	4200	Operational	CLR Enabled	The Common Language Runtime (CLR) configuration option is enabled for the SQL Server instance. You can run managed code in the .NET Framework.
50	4210	Operational	OLE Automation Disabled	The OLE Automation Procedures configuration option is disabled (2005) or the procedures were dropped (2000) for the SQL Server instance. SQL diagnostic manager does not use OLE automation to collect data such as OS metrics and service details.
51	4220	Queries	Query Monitor Events (Count)	The number of queries captured by the Query Monitor on the SQL Server instance during the last scheduled refresh.
52	4230	Operational	SQL diagnostic manager Collection Service Status	Status of the SQL diagnostic manager Collection Service.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
53	4240	Operational	SQL Server Agent XPs Disabled	The Agent XPs configuration option is disabled (2005) or the procedures were dropped (2000) for the SQL Server instance. SQL diagnostic manager does not collect SQL Agent job details.
54	4250	Operational	WMI Service Unavailable	The WMI service is unavailable on the SQL Server computer. SQL diagnostic manager does not collect OS metrics.
57	4310	Sessions	Client Computers (Count)	Number of unique client computers connected to the SQL Server instance.
58	4320	Sessions	Blocked Sessions (Count)	Number of sessions on the SQL Server instance blocked by other sessions holding requested locks.
59	4330	Operational	SQL Server Data Used (Percent)	Percentage of allocated data space used across all data files on the SQL Server instance.
60	4340	Operational	SQL Server Log Used (Percent)	Percentage of allocated log space used across all log files on the SQL Server instance.
62	4110	Resources	OS Disk Time Per Disk (Percent)	Percentage of elapsed time that an individual disk was busy servicing read and write requests on the SQL Server computer. This event alerts separately for each disk on a server.
63	4120	Resources	OS Average Disk Queue Length Per Disk (Count)	Average number of both read and write requests queued for an individual disk on the SQL Server computer. This event alerts separately for each disk on a server.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
64	4100	Resources	OS Disk Full (Percent)	Percentage of space used on a logical disk on the SQL Server computer. This event alerts separately for each disk on a server.
65	2940	Services	SQL Server Agent Long Running Job (Minutes)	Maximum number of minutes a scheduled SQL Server Agent job should take to complete.
66	4400	Logs	SQL Server Error Log	The number of error log events requiring attention based on either message severity or specified text.
67	4410	Logs	SQL Server Agent Log	The number of agent log events requiring attention based on either message severity or specified text. When configuring this alert, use the indicator or grid to specify which log message severity corresponds to warning or critical. Use the advanced options to specify additional filters.
68	5105	Databases	Mirroring Unsent Log (KB)	Size of the unsent log in the send queue on the principal in kilobytes.
69	5106	Databases	Mirroring Unrestored Log (KB)	Size of the redo queue on the mirror in kilobytes.
70	5107	Databases	Mirroring Oldest Unsent Transaction (Minutes)	The age of the oldest unsent transaction in minutes on the principal. This is only meaningful on the principal server instance.
71	5108	Databases	Mirror Commit Overhead (Milliseconds)	Average delay (in milliseconds) on the principal server instance for each transaction because of database mirroring.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
72	5109	Databases	Mirroring Status	Status of the mirrored database such as synchronized suspended or disconnected.
73	5110	Databases	Mirroring Preferred Configuration	This operational configuration of this mirroring session changed.
74	5111	Databases	Mirrored Server Role Change	A mirrored database role changed.
75	5112	Databases	Mirroring Witness Connection	The mirroring witness is not connected.
76	4130	Resources	Page Life Expectancy	The number of seconds a page stays in the buffer pool without references.
77	4500	Services	Cluster Failover	A cluster failover has occurred.
78	4510	Services	Cluster Active Node	The active node of the cluster is a non-preferred node. The preferred node may be configured under Server Properties>Cluster Settings.
79	5120	Services	Distribution Latency (Seconds)	Time in seconds that a replication transaction has been holding at the Publisher and waiting to be received by a Distributor.
80	2840	Sessions	Deadlock	A deadlock occurred on the monitored server. Note that the Activity Monitor >Capture deadlocks (SQL 2005+) check box must be enabled for this alert to function.
81	4140	Resources	Procedure Cache Hit Ratio	The percentage of procedure cache hits to cache lookups.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
82	4150	Resources	Average Disk Milliseconds Per Read	Average time in milliseconds of a read operation of data from the disk. This is a per-disk metric.
83	4160	Resources	Average Disk Milliseconds Per Transfer	Average time in milliseconds of a disk transfer operation. This is a per-disk metric.
84	4170	Resources	Average Disk Milliseconds Per Write	Average time in milliseconds of a write operation of data to the disk. This is a per-disk metric.
85	4180	Resources	Disk Reads Per Second	Rate of read operations on the disk. This is a per-disk metric.
86	4185	Resources	Disk Transfers Per Second	Rate of transfer operations on the disk. This is a per-disk metric.
87	4190	Resources	Disk Writes Per Second	Rate of write operations on the disk. This is a per-disk metric.
88	3140	Services	SQL Server Agent Job Completion	The SQL Server Agent job/step has completed.
89	5130	Databases	Version Store Generation Ratio	Amount in percent by which the tempdb version store generation rate exceeds the cleanup rate. This alert is only raised on instances running SQL Server 2005 or above.
90	5131	Databases	Version Store Size (MB)	Size in megabytes of the tempdb version. This alert is only raised on instances running SQL Server 2005 or above.
91	5135	Databases	Longest Running Version Store Transaction (Minutes)	Time in minutes that the longest version store transaction has been opened and preventing version store cleanup. This alert may only be raised on instances running SQL Server 2005 or above.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
92	5136	Databases	Session Tempdb Space Usage (MB)	Amount of tempdb space being used by a running session. This alert is only raised on instances running SQL Server 2005 or above.
93	5132	Databases	Tempdb Contention (ms)	Current wait time for tempdb allocation maps (GAM SGAM and PFS) in milliseconds. This alert is only raised on instances running SQL Server 2005 or above.
94	1115	Databases	Log File Autogrow	Log file autogrow occurred on the specified database. Enable the "Activity Monitor>Capture Autogrow check box for this alert to function.
95	1116	Databases	Data File Autogrow	Data file autogrow occurred on the specified database. Enable the Activity Monitor>Capture Autogrow check box for this alert to function.
96	5200	Virtualization	VM Resource Configuration Change	A resource configuration change occurred for the virtual machine hosting the monitored SQL Server instance. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.
97	5201	Virtualization	VM Host Server Change	The virtual machine hosting the monitored SQL Server instance moved to a different host. This alert is disabled by default and is only relevant to instances hosted on VMWare vCenter which have been enabled for VM monitoring in the SQLdm VM Configuration window.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
98	5202	Virtualization	VM CPU Usage (Percent)	The percentage of available virtual CPU time used by the virtual machine hosting the monitored SQL Server. This may differ from the processor utilization percentage reported by the guest OS. A high percentage may indicate a large number of active client sessions. This alert is disabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.
99	5250	Virtualization	Host CPU Usage (Percent)	The percentage of available CPU time used by the host server hosting the virtual machine on which the monitored SQL Server is currently running. This alert is enabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.
100	5203	Virtualization	VM Memory Usage (Percent)	The percentage of available memory used by the virtual machine hosting the monitored SQL Server. This may differ from the memory usage percentage reported by the guest OS. This alert is disabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
101	5251	Virtualization	Host Memory Usage (Percent)	The percentage of available memory used by the host server hosting the virtual machine on which the monitored SQL Server is currently running. This alert is enabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.
102	5204	Virtualization	VM CPU Ready Wait Time	The time in milliseconds that a virtual CPU has been waiting for a physical CPU to become available for processing for the virtual machine on which the monitored SQL Server is currently running. A spike in this value may indicate CPU pressure on the host server. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.
103	5205	Virtualization	VM Reclaimed/ Ballooned Memory (KB)	The amount of memory that has been reclaimed by host and is no longer available to the virtual machine on which the monitored SQL Server instance is running. A spike in this value may be a sign of memory pressure on the host. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
104	5206	Virtualization	VM Memory Swap Delay Detected	Memory being swapped to disk is causing a performance delay on the virtual machine on which the monitored SQL Server instance is running. This is triggered if the Memory Swap In Rate (KB/s) or CPU Swap Rate (ms) counters are non-zero. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.
105	5252	Virtualization	Host Memory Swap Detected	Memory is being swapped to disk for the host server hosting the virtual machine on which the monitored SQL Server is currently running. This may be an indication of memory pressure on the host. This alert is disabled by default and is only relevant to instances hosted on VMWare vCenter which have been enabled for VM monitoring in the SQLdm VM Configuration window.
106	5207	Virtualization	VM Resource Limits Detected	Either CPU or memory resource limits are set for the virtual machine hosting the monitored SQL Server instance. Resource limits are soft limits set by the VM Administrator on the total amount of resources that a VM is allowed to consume. This alert is disabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
107	5208	Virtualization	VM Power State	This is the state of the VM as reported by the vCenter or Hyper-V host servers. This alert is enabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.
108	5253	Virtualization	Host Power State	This is the state of the host as reported by the vCenter or Hyper-V host servers. This alert is enabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.
109	1113	Databases	Database Full (Size)	The amount of disk space (in MB) currently used by the database as the sum of data text and indexes. This value is the actual amount of used data within the database file not the size of the database file itself.
110	1214	Databases	Log Full (Size)	The amount of disk space (in MB) currently used by the transaction log for this database. This value is the actual amount of data within the log not the size of the database log file itself.
111	4101	Resources	OS Disk Free Space (Size)	Amount of space (in MB) available on a logical disk on the SQL Server computer.
116	5301	Databases	Availability Group Role Change	Primary role changed to a secondary replica. Alert includes affected database(s) for the entire availability group.

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
117	5302	Databases	Availability Group Estimated Data Loss (Seconds)	The time period in seconds that data is not synchronized to the secondary replica. Value reflects the time difference of the last commit between the primary and secondary replicas. Alert includes affected database(s).
118	5303	Databases	Availability Group Synchronization Health	Reflects database synchronization state in the availability group. Possible values include: not healthy partially healthy and healthy.
119	5304	Databases	Availability Group Estimated Recovery Time (Seconds)	The required time in seconds that the secondary replica needs to catch up with the primary replica. Alert includes affected database(s).
120	5305	Databases	Availability Group Synchronization Performance (Seconds)	The time required in seconds to complete synchronization between the primary and secondary replicas.
121	5306	Databases	Availability Group Log Send Queue Size (KB)	Amount of log records in KB needed to ship to the secondary replica to complete synchronization. Alert includes affected database(s).
122	5307	Databases	Availability Group Redo Queue Size (KB)	Amount of log records from log files in KB that need redoing in the secondary replica to complete synchronization. Alert includes affected database(s).
123	5308	Databases	Availability Group Redo Rate (KB/s)	Rate in KB per second at which log records are redone in the secondary database to complete synchronization. Alert includes affected database(s).

Repository ID	Windows Event ID	Category	Metric Name	Metric Description
124	5320	Databases	Availability Group Preferred Replica	Alert is raised when the database is no longer on the preferred replica.
125	5324	Operational	SQLdm Repository Grooming Timed Out	Grooming timed out on your SQL Server Repository. SQL diagnostic manager no longer deletes stored metrics in the SQLdm Repository at the set schedule. To review your grooming settings access Tools > Grooming Options.
126	5328	Databases	Filegroup Space Full (Percent)	TBD
127	5330	Databases	Filegroup Space Full (Size)	TBD
128	5332	Services	SQL Server Browser Service Status	Status of the SQL Server Browser service.
129	5342	Services	SQL Server Active Directory Helper Service Status	Status of the SQL Server Active Directory Helper service.
130	1115	Backup	Days since last backup	Days since the database has not been backed up.

Troubleshoot WMI connectivity issues

In order to use WMI, the user account under which the Collection Service runs must have administrator permissions on the remote server. The most frequently encountered problems with WMI relate to RPC traffic failing to get through to the remote computer. Another issue involves DCOM/WMI permissions.

The following links provide additional information about how to troubleshoot WMI connectivity issues:

- [Securing a remote WMI connection](#)
- [Help with Scripts](#)

Using WbemTest (Windows Management Instrumentation Tester)

You can use the WbemTest tool to connect to a server and issue WMI queries. Download this tool from Microsoft TechNet to help you test and troubleshoot WMI issues.

To use WbemTest:

1. Run `wbemtest.exe`.
2. Click **Connect**.
3. In the **NameSpace text** box, enter `\\server\root\cimv2` where server is the name of the server to which you want to connect.
4. Provide Username and Password.
5. Click **Connect**.
6. Click **Query**.
7. Enter `select * from win32_process`.
8. Click **Apply**.

If WbemTest connects to the remote server and issues the query using WMI, you should see a query result with output. In this case, WMI to the required server is working and no further action is needed. For more information on the Windows Management Instrumentation Tester, refer to the Microsoft document, [Windows Management Instrumentation Tester overview](#).

If you receive an error message, use the following processes to help identify and resolve the issue.

Error: The RPC Server Is Unavailable

This error usually indicates that the RPC traffic is not getting to the remote server, or there is no RPC listener on the remote server.

To troubleshoot this RPC error:

1. Make sure that the Remote Procedure Call (RPC) service is running on the remote server.
2. Verify that there is a TCP listener on the remote server by running the `netstat -nao` command and verifying that there is the following entry: `TCP 0.0.0.0:135 0.0.0.0:0 LISTENING 1304`
3. In the Tools subdirectory, run `rpcping /s <servername> /t ncacn_ip_tcp` where `<servername>` is the name of the remote server. This command verifies that RPC can communicate with the remote server and output similar to:

```
Completed 1 calls in 15 ms
66 T/S or 15.000 ms/T
```
4. Make sure that local or internal network firewalls do not block traffic by either disabling the firewall or configuring the Windows firewall to allow incoming RPC traffic.
5. Try to use the remote server IP address instead of the server name. If the IP address works, you may have a DNS issue.

6. If the remote server resides in a different domain, the two domains may not trust each other, or the user account does not have administrator permissions on the remote server/domain.
7. If both computers are in the same domain, and the user account has administrator permissions, try rejoining both computers to the domain.

Error: Access Denied

This error can indicate permission issues.

To troubleshoot this access error:

1. If the remote computer is running Windows XP, make sure it is not set to Force Guest. This setting forces impersonation of any connection as Guest.
 - a. Open the Local Security Policy console from Administrative Tools.
 - b. Browse to **Security Settings > Local Policies > Security Options**.
 - c. Double-click Network Access: Sharing And Security Model For LocalAccounts.
 - d. Change the settings from **Guest Only** to **Classic**.
2. Make sure DCOM is enabled on the remote server:
 - a. Run DcomCnfg on the remote server.
 - b. Click **Component Services**.
 - c. Expand **Computers**.
 - d. Right click **My Computer** and select **Properties**.
 - e. Click the **Default Properties** tab.
 - f. Make sure **Enable Distributed COM** on this computer is checked.
3. Verify the configuration of the correct DCOM remote launch and activation permissions:
 - a. Run DcomCnfg on the remote server.
 - b. Click **Component Services**.
 - c. Expand **Computers**.
 - d. Right click **My Computer** and select **Properties**.
 - e. Click the **Com Security** tab.
 - f. Under **Launch and Activation Permissions**, click **Edit Limits**.
 - g. In the Launch Permissions dialog box, make sure your user account or group is listed in the Groups or user names list. If your user account or group is not listed, click **Add** and add it to the list.
 - h. In the Launch Permission dialog box, select your user account or group in the Group or user names list. In the Allow column under Permissions for User, select **Remote Launch** and **Remote Activation**, and then click **OK**.
4. Make sure the correct DCOM remote access permissions are configured:
 - a. Run DcomCnfg on the remote server.
 - b. Click **Component Services**.
 - c. Expand **Computers**.
 - d. Right click **My Computer** and select **Properties**.
 - e. Click the **Com Security** tab.
 - f. Under **Access Permissions**, click **Edit Limits**.
 - g. In the Access Permission dialog box, select ANONYMOUS LOGON name in the Group or user names list. In the Allow column under Permissions for User, select **Remote Access**, and then click **OK**.
5. Make sure the correct WMI namespace permissions are configured.
 - a. On the target machine, run wmi mgmt . msc.
 - b. Right-click **WMI Control**, and then select **Properties**.
 - c. In the Security tab, select the CIMV2 namespace, and then click **Security**.
 - d. Locate the appropriate account, and then check **Remote Enable** in the Permissions list.


Integrate SQL Diagnostic Manager with SCOM

The IDERA SQLDM Management Pack (SQLDM MP) integrates and extends the Microsoft System Center Operations Manager (SCOM) monitoring capabilities by providing database administrators and IT operators with additional insight into the availability and performance of their SQL Server interfaces, all from one console.

SQLDM MP includes automatic discovery of SQLDM installations and of the SQL Server instances managed from those installations. Once discovered, SQLDM MP propagates the current status, health, and events of monitored instances up to SCOM.

SQLDM uses [Alert Actions and Responses](#) to control which alerts are escalated to SCOM and whether they go into SCOM as an Event or an Alert. For additional information, go to [SCOM Event Action](#) or [SCOM Alert Action](#).

For additional information about SQLDM Management Pack, click [Product Documentation](#), select the SQL Diagnostic Manager version you need, and click View. A dialog box opens with documentation options for SQL Diagnostic Manager plug-ins, click System Center Management Pack option, and download the SQLDM MP file.

 Note that the IDERA SQLDM Management Pack for integration with SCOM is designed for System Center Operations Manager 2007 R2 and above.

Understanding the SQLDM Management Pack health model

The SQLDM Management Pack reports the health of each SQL Server instance monitored by SQL Diagnostic Manager, as well as the server health of SQL Diagnostic Manager itself. The following main characteristics help define the SQL Diagnostic Manager health model:

SQLDM Server

The health of SQL Diagnostic Manager is tracked by monitoring the state of three underlying services, using the following three service monitors:

- Collection Service Monitor
- Management Service Monitor
- Predictive Service Monitor

The automatically discovered Object ID for the SQLDM Service is Idera.SQLDM.Service.

SQLDM Instance

The health of each SQL server instance being monitored by SQL Diagnostic Manager is rolled up by reflecting the Alerts counts opened for that instance within SQL Diagnostic Manager.

If a SQL server instance currently has one or more Critical alerts open, then the Health of that instance is recorded as Critical. If one or more Warning alerts are open, but no Critical alerts show, then the health of the instance is recorded as Warning.

In Health Explorer the Diagnostic Manager Alert status which drove a specific Health state change is recorded and displayed in the State Change Events tab.

The automatically discovered Object ID for a SQL Instance is Idera.SQLDM.SqlInstance.

Install and configure the SQLDM Management Pack for SCOM

Before importing the SQLDM Management Pack for SCOM, you must import the relevant Microsoft Windows management packs for your current operating system. For example, if SQL Diagnostic Manager is installed on Windows Server 2008 R2, then you must install the management pack for that OS before continuing.

You can find the SQLDM MP (Idera.SQLDM.mp) in the root of the installation kit.

Configuring your installation to support SCOM

The SQLDM Management Pack requires that the SCOM agent has at least read access to the SQLDM Repository. Grant this access before installing the SQLDM MP and enabling discovery of any SQL Diagnostic Manager installed instances.

SQLDM MP also requires that a SCOM agent (version 2007 or above) be installed locally on each server where an instance of SQL Diagnostic Manager is present. If an agent is not already installed, then use the Discovery Wizard in the SCOM Console to push the agent installation.

Importing the SQLDM Management Pack for SCOM

The SQLDM Management Pack import process loads into the SCOM database the SQLDM MP settings, classes, and rules necessary for monitoring SQL Diagnostic Manager. The import replicates the relevant rules and objects to all systems where SQL Diagnostic Manager is installed.

The following steps help you import the SQLDM Management Pack when it is downloaded to a hard drive local to the SCOM Console. For more information about other methods of importing the Management Pack, see [How to Import a Management Pack in Operations Manager 2007](#).

To import the SQLDM Management Pack:

1. Open the Operations Console for Operations Manager, and then click the Administration button.
2. In the Actions pane or from Actions → Actions menu option, click Import Management Packs.
3. Click **Add**, and then click **Add from disk**.
4. In the Online Catalog Connection dialog, click **No**. **If you did not preload the required OS Management Packs and you have a current Internet connection**, click **Yes**.
5. Browse for and then select the IDERA SQL Diagnostic Manager Management Pack.
6. Click **Install**. Discovery of SQL Diagnostic Manager servers and monitored SQL Server instances begins automatically as the Management Pack is deployed to the relevant agents across the network.

View SQL Diagnostic Manager details in SCOM

With the SQLDM Management Pack, all the key monitors employed by SQL DBAs are now fully visible to System Center Operations Manager operators, synchronizing IT operations and DBA functions and identifying situations that could impact SQL Server availability and performance.

The following table lists the SQL Diagnostic Manager views available within SCOM:

View Name	Description
SQL DM Server Health	This Health view lists all discovered SQL Diagnostic Manager servers in the environment. The health state of each instance is displayed, together with information about the specific SQL Server instance.
SQL Instance Alerts	Displays a list of current Alerts generated when a specific SQL Server instance monitored by SQL Diagnostic Manager has entered an unhealthy state.
SQL Instance Events	This view offers a list of all Alert transitions recorded by SQL Diagnostic Manager for each SQL Server instance which is being monitored. This view provides a complete history of the status of all monitored SQL Server instances.
SQL Instance Health	The Health view displays the current health state of each SQL Server instance monitored by a SQL Diagnostic Manager server.
Dashboard	This view combines the SQL DM Server Health, SQL Instance Health and the SQL Server instance events views into a single pane.

To access the SQL Diagnostic Manager views available in SCOM:

1. In the Monitoring pane of the System Center Operations Manager Console, locate the IDERA SQLDM folder.
2. Expand the tree view by clicking + sign next to **SQLDM**.